

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

DESENVOLVIMENTO E APLICAÇÃO DE UM SISTEMA DE VISÃO PARA ROBÔ
INDUSTRIAL DE MANIPULAÇÃO

por

Maurício Velloso Grassi

Dissertação para obtenção do Título de
Mestre em Engenharia

Porto Alegre, maio de 2005

DESENVOLVIMENTO E APLICAÇÃO DE UM SISTEMA DE VISÃO PARA ROBÔ
INDUSTRIAL DE MANIPULAÇÃO

por

Maurício Velloso Grassi

Engenheiro Mecânico

Dissertação submetida ao Corpo Docente do Programa de Pós-Graduação em Engenharia Mecânica, PROMEC, da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como parte dos requisitos necessários para a obtenção do Título de

Mestre em Engenharia

Área de Concentração: Processos de Fabricação

Orientador: Prof. Dr. Flávio José Lorini

Aprovada por:

Prof. Dr. Eduardo André Perondi, PROMEC/UFRGS

Prof. Dr. Ney Francisco Ferreira, DEMEC/UFRGS

Prof. Dr. Gilberto Dias da Cunha, PUCRS

Prof. Dr. Flávio José Lorini

Coordenador do PROMEC

Porto Alegre, maio de 2005.

AGRADECIMENTOS

Ao orientador, pelo apoio e motivação.

Aos colegas, pelo auxílio e companheirismo.

À comissão de avaliação, pelas críticas e sugestões.

À esposa e família, pelo carinho e atenção.

RESUMO

O presente trabalho aborda o desenvolvimento e aplicação de um sistema de visão para um robô industrial, dedicado a tarefas de manipulação. Para seu desenvolvimento, foi feita a integração de equipamentos comerciais de captação de vídeo com um PC, onde é executada a análise das imagens. Os resultados são enviados ao controlador do robô através de um sistema de comunicação, sendo recebidos por um programa de manipulação que executa as tarefas.

Como peça central do sistema de captura e processamento de imagens tem-se o programa RobVis, desenvolvido em linguagem *Visual Basic* e que tem a função principal de identificar a posição e orientação de objetos a serem manipulados pelo robô, que interpreta as imagens captadas por uma câmera de vídeo do tipo CCD, ligada a uma placa de captura de imagens. Após definidas as variáveis de interesse, estas são transmitidas ao controlador do robô, através de um sistema de comunicação IRBCom, desenvolvido no Laboratório de Robótica da UFRGS. No controlador, um programa de manipulação escrito em linguagem *Rapid*, nativa do manipulador empregado, recebe as variáveis para execução da tarefa de captura de objetos em posições aleatórias e depósito em um ponto de descarga pré-definido.

O sistema de visão desenvolvido caracteriza-se como de fácil implementação e aplicação em tarefas de manipulação robótica industrial que exijam a determinação da posição e orientação de objetos de trabalho dentro de uma porção do volume de trabalho do robô, coberta por um sistema de visão.

ABSTRACT

“Development and Application of a Vision System for Manipulating Robot”

This work deals with the development and application of a vision system for an industrial robot dedicated to manipulating tasks. For its development, a commercial video-capture equipment was combined to a PC, where image analysis is performed. The results are sent to the robot controller through a communication system and received by a manipulating program which performs the tasks.

The main piece of the image capture and processing system is the processing software RobVis, developed using the Visual Basic code, that has the main function of identify the position and orientation of objects intended to be manipulated by the robot. The software processes the images captured by a CCD video camera connected to an image capture card. When the desired variables are calculated, these are transmitted to the robot controller through an IRBCom communication system, developed by the UFRGS Laboratory of Robotics. At the controller, a manipulating program written in Rapid, the native language of the robot, waits for the variables to perform the task of capturing parts in randomic positions and depositing them at a pre-established unloading point.

The designed vision system is qualified as of easy usage and application in robotic manipulating tasks that need information about position and orientation of parts inside a section of the robot's work space that is within the in range of the vision system.

ÍNDICE

ÍNDICE DE TABELAS	x
1. Introdução.....	1
1.1 Motivação	2
1.2 Objetivos e Limitações do Trabalho.....	2
2. Revisão Bibliográfica	4
2.1. O Robô.....	4
2.2 Aplicações dos Robôs.....	6
2.3 Visão Robótica	8
2.3.1 Imagem	9
2.4 Captação da Imagem	11
2.4.1 Câmera Vidicon	11
2.4.2 Câmera Digital.....	13
2.5 Técnicas de Iluminação	15
2.6 Pré-Processamento de Imagens	16
2.6.1 Domínio-Frequência	16
2.6.2 Domínio-Espaço	18
2.6.3 Conectividade	20
2.7 Processamento de Imagens.....	21
2.7.1 Histograma da Imagem.....	22
2.7.2 Limiarização	23
2.7.3 Detecção da Borda.....	24
2.7.4 Segmentação Orientada a Regiões.....	27
2.7.5 Operações de Morfologia Binária.....	29
2.8 Análise de Imagens.....	31
2.8.1 Reconhecimento de Objetos	31
2.8.2 Momentos	33
2.9 Sistemas de Controle e Transmissão de Sinais.....	36
2.9.1 Hierarquia de Sistemas de Automação Industrial.....	37
2.9.2 Redes Locais	38
3. Desenvolvimento do Sistema	40
3.1 Equipamento de Captura de Imagens	40
3.1.1 Câmera.....	40

3.1.2 Placa de Captura	41
3.1.3 Suporte da Câmera.....	42
3.2 Iluminação	43
3.3 Interface PC/Controlador do Robô.....	44
3.3.1 Módulo Hardware IRBCom	46
3.3.2 Módulo Software IRBCom.....	47
3.4 Programa de Análise de Imagens	48
3.4.1 Módulo de captura	48
3.4.2 Módulo de Processamento	49
3.4.3 Módulo de Calibração.....	52
3.4.4 Procedimentos Matemáticos	53
3.5 Programa de Manipulação	57
4. Avaliação Experimental	60
4.1 Desenvolvimento da Tabela de Comparação	61
4.2 Manipulação de Objetos com Orientação.....	62
4.2.1 Carregando o Programa de Manipulação	63
4.2.2 Referência do Campo de Visão	63
4.3 Seleção de Parâmetros	64
4.4 Ciclo de Operação	65
4.5 Operação com Sólido de Revolução.....	66
5. Considerações Finais.....	68
5.1 Resultados.....	68
5.2 Conclusões.....	69
5.3 Sugestões	69
Referências Bibliográficas	71

LISTA DE SÍMBOLOS

AC	<i>Alternate Current</i> (Corrente Alternada)
BMP	<i>Bit MaP</i> (Mapa de Bits)
BARA	<i>British Automation and Robot Association</i> (Associação Britânica de Robótica e Automação)
CAD	<i>Computer Aided Design</i> (Projeto Assistido por Computador)
CAM	<i>Computer Aided Manufacture</i> (Fabricação Assistida por Computador)
CCD	<i>Charge-Coupled Device</i> (Dispositivo de carga Acoplada)
CFTV	Circuito Fechado de Televisão
CID	<i>Charge-Integrated Device</i> (Dispositivo de Carga Integrada)
CIM	<i>Computer Integrated Manufacturing</i> (Manufatura Integrada por Computador)
CNC	<i>Computerized Numerical Control</i> (Controle Numérico Computadorizado)
CRT	<i>Cathode-Ray Tube</i> (Tubo de Raios Catódicos)
DLL	<i>Dinamic Link Library</i> (Biblioteca de Vínculo Dinâmico)
EIA	<i>American Electronic Industry Association</i> (Associação Americana da Indústria Eletrônica)
FFT	<i>Fast Fourier Transform</i> (Transformada Rápida de Fourier)
FM	<i>Frequency Modulated</i> (Modulado em Frequência)
fps	<i>Frames per Second</i> (Quadros por Segundo)
GPFAI	Grupo de Projeto, Fabricação e Automação Industrial
IFTtoMM	<i>International Federation for the Promotion of Mechanism and Machine Science</i> (Federação Internacional para a Promoção da Ciência de Mecanismos e Máquinas)
ISO	<i>International Standards Organization</i> (Organização Internacional de Padrões)
JPEG	<i>Joint Photographic Experts Group</i> (Grupo Combinado de Peritos em Fotografia)
LAN	<i>Local Area Network</i> (Rede Local)
NASA	<i>National Aeronautics and Space Administration</i> (Administração Nacional de Aeronáutica e Espaço)
NTS	<i>National Television Standards</i> (Padrões de Televisão Nacionais)
OSI	<i>Open System Interconnect</i> (Interconexão de Sistema Aberto)
PC	<i>Personal Computer</i> (Computador Pessoal)
PCI	<i>Peripheral Component Interconnect</i> (Interconexão para Componente Periférico)
RGB	<i>Red, Green, Blue</i> (Vermelho, Verde, Azul)

RIA	<i>Robot Industry Association</i> (Associação da Indústria Robótica)
SCARA	<i>Selective Compliance Assembly Robot Arm</i> (Braço Robótico Seletivo de Montagem)
TIFF	<i>Tagged Image File Format</i> (Formato de Arquivo de Imagem Rotulada)
USB	<i>Universal Serial Bus</i> (Barra Serial Universal)

ÍNDICE DE FIGURAS

Figura 2.1 – Tipos de robôs.....	5
Figura 2.2 – Esquema de uma câmera Vidicon.....	12
Figura 2.3 – Varredura da câmera Vidicon.....	12
Figura 2.4 – Modelo de captura de dados de imagem.....	14
Figura 2.5 – Formas de iluminação.....	16
Figura 2.6 – Vizinhança 3 × 3 de um ponto da imagem e detalhe da máscara de convolução.....	19
Figura 2.7 – Máscara 3 × 3 mostrando coeficientes e coordenadas de <i>pixel</i>	20
Figura 2.8 – Vizinhança de <i>pixel p</i>	21
Figura 2.9 – Histogramas de 4 tipos básicos de imagens.....	23
Figura 2.10 – Processo de limiarização.....	24
Figura 2.11 – Detecção da borda por operadores de derivação.	25
Figura 2.12 – Máscaras de Sobel.	27
Figura 2.13 – Efeito aproximado de espessamento (a) e dilatação (b).	30
Figura 2.14 – Erosão (a) e esqueletonização sem (b) e com (c) espessamento prévio.	31
Figura 2.15 – Razão de Aspecto de um objeto.....	32
Figura 2.16 – Imagem exemplo para cálculo de momentos.....	34
Figura 2.17 – Objetos com pequenas diferenças e assimetrias.	35
Figura 3.1 – Câmera CCD.....	41
Figura 3.2 – Placa de captura de imagens.	42
Figura 3.3 – Suporte da câmera.....	43
Figura 3.4 – Iluminação da área de captura.	44
Figura 3.5 – Sistema segundo o modelo de camadas <i>OSI</i>	46
Figura 3.6 – Módulo IRBCom.	46
Figura 3.7 – Tela do módulo de captura com detalhes.....	49
Figura 3.8 – Barra de menu do módulo de processamento.	50
Figura 3.9 – Interface do módulo de processamento, com observações para suas funções.....	52
Figura 3.10 – Interface do módulo de calibração, com observações para suas funções.	53
Figura 3.11 – Área de captura com os ângulos para calibração.....	57
Figura 3.12 – Comunicação entre os componentes do sistema.....	57
Figura 3.13 – Trajetória seguida pelo manipulador.	59
Figura 4.1 – Ferramentas e corpos de prova.	60
Figura 4.2 – Aparência do módulo e imagem durante calibração para 45°.....	61

Figura 4.3 – Carregando base de dados.....	62
Figura 4.4 – Procedimento de referenciar o campo de visão.	64
Figura 4.5 – Seleção dos parâmetros.....	65
Figura 4.6 – Imagens captadas durante o ciclo de operação.	66
Figura 4.7 – Imagens de manipulação de sólido de revolução.....	67

ÍNDICE DE TABELAS

Tabela 3.1 – Dados técnicos do conjunto.....	41
Tabela 4.1 – Tabelas de comparação para seção “T” e retangular.....	62
Tabela 5.1 – Resultados dos testes.....	68

1. INTRODUÇÃO

Há não muito tempo, surgiu no imaginário da humanidade, algo que pudesse facilitar o trabalho das pessoas e até substituí-las na execução de tarefas repetitivas e difíceis. Pode-se dizer que o surgimento da robótica está ligado ao livro tcheco *Robôs Universais de Rossum* [Čapek apud Niku, 2001]. Nesta obra, os *Robotas* (trabalhadores), máquinas semelhantes aos humanos, seriam produzidos em larga escala e empregados nas mais variadas tarefas. Dos *Robotas*, originou-se o termo robô e derivou a expressão robótica que é a base científica e conhecimento para se desenvolver, aplicar e usar robôs em tarefas essencialmente humanas.

Saindo da ficção, o emprego da robótica no setor industrial vem desde a década de 1960, após as experiências desenvolvidas na década anterior, onde foram construídos os primeiros robôs industriais, dos quais ressaltaram-se as principais vantagens do seu emprego:

- a automação pode, em várias situações, aumentar a produtividade, segurança, eficiência e qualidade dos produtos;
- os robôs podem trabalhar em ambientes perigosos sem a necessidade de suporte de vida, conforto ou preocupações com segurança;
- os robôs não precisam de conforto ambiental, como iluminação, ar condicionado, ventilação e proteção de ruídos;
- os robôs trabalham continuamente sem apresentar fadiga e cansaço;
- robôs apresentam repetitividade de precisão durante todo o ciclo de operação;
- os robôs podem ser muito mais precisos que os humanos.

Mas o uso da robótica também apresenta algumas deficiências, dentre elas podem ser citadas:

- os robôs não têm a habilidade de responder a emergências, a menos que a situação seja prevista e a resposta esteja incluída em seu sistema. Medidas de segurança são necessárias para garantir que não sejam causados danos a operadores e máquinas trabalhando junto à eles [Bonney e Yong, 1985]. Isto inclui respostas inapropriadas ou erradas, falta de poder de tomada de decisão, perda de potência, danos ao robô e outros dispositivos e lesões a humanos;
- robôs são relativamente caros, devido ao custo inicial do equipamento, custos de instalação, necessidade de periféricos, operadores treinados e de programação específica;

- os robôs, apesar de serem equipamentos com vasta gama de aplicações, podem apresentar deficiências em alguns casos devido a limitadas capacidades em graus de liberdade, sensores integrados e resposta em tempo-real.

Este último aspecto, mais precisamente a falta de um sistema de visão integrado e a capacidade de envio de decisões em tempo-real serão o foco principal deste trabalho.

1.1 Motivação

Apesar do emprego de robôs industriais estar plenamente estabelecido em tarefas repetitivas e que dependam apenas de um programa pré-concebido, que lida com parâmetros fixos, o mesmo não se pode dizer do campo em que se tratam parâmetros variáveis. Necessita-se, neste caso, de uma programação que se auto-adapte para que a vantagem da não-necessidade de constante intervenção humana em tarefas industriais desapareça.

Uma maneira para se lidar com esta dificuldade surge através do emprego de sistemas de visão, capazes de captar o ambiente de trabalho, mais especificamente a área em torno do foco central do processo do robô. É necessário que o sistema seja capaz de interpretar esta imagem e transmitir um comando para que o robô execute a tarefa adaptando-se a condições operacionais. Necessita-se, neste caso, também de um sistema de comunicação eficiente, que transmita as informações ao robô, para que este possa responder de imediato às variações.

Verificando esta peculiaridade de emprego a que alguns sistemas robóticos são submetidos, optou-se pela abordagem deste tema, visando desenvolver e aplicar um sistema de visão em tarefas de manipulação. Para tanto, busca-se a integração de equipamentos de captação de vídeo encontrados no mercado com um robô industrial, através de um sistema de comunicação utilizando um computador pessoal para o processamento da imagem.

1.2 Objetivos e Limitações do Trabalho

O objetivo principal do trabalho foi desenvolver um sistema de visão com as seguintes características:

- De construção simples a partir de componentes de tecnologia consolidada: em termos de empregar equipamentos disponíveis no mercado e suficientemente robustos para aplicações na automação de sistemas industriais;
- De baixo custo: comparado à sistemas fechados comerciais, compreendendo equipamento de captura de imagem, análise e envio de sinais;

- De fácil implementação: tanto em termos de integração e compatibilidade de equipamentos quanto em simplicidade de operação;
- Capaz de ser empregado em tarefas de manipulação robótica industrial: tarefas que necessitam informação de posição e orientação de componentes em diferentes condições de captação da imagem, com eficiente envio dos dados resultantes de sua análise ao robô.

Sistemas de visão com as características citadas encontram uma ampla gama de aplicações na automatização industrial tais como embalagem de produtos manufaturados, paletização e movimentação e posicionamento de componentes entre estações de trabalho dentro de uma mesma célula de fabricação.

A proposta do trabalho limita-se a identificação da posição de componentes e sua orientação dentro de um espaço de trabalho bem definido e calibrado para operações específicas. Nestas condições é suficiente, para a extração das informações necessárias, a análise de imagens bidimensionais, ou seja, sobre um plano (x,y) , com o emprego de apenas uma câmera. Limita-se, portanto, à verificação da posição e orientação de objetos neste plano, não extraindo informação de altura (cota z), peculiar de sistema de visão estéreo, que empregam duas, ou mais, câmeras. Não se buscou, também, a identificação e diferenciação de objetos na cena captada, estando o sistema, limitado a analisar imagens contendo apenas um objeto.

2. REVISÃO BIBLIOGRÁFICA

Tendo em vista o foco deste trabalho, buscou-se fazer uma breve apresentação dos pontos de interesse utilizados em seu desenvolvimento, tais quais: definição e descrição dos robôs industriais, introdução à visão industrial e seus equipamentos, técnicas de processamento e análise de imagens e transmissão de sinais. Cabe ressaltar que há um grande número de técnicas empregadas para visão industrial, sendo aqui apresentadas apenas as de aplicação mais comum e de certo grau de facilidade de implementação.

2.1. O Robô

Para qualificar o que é um robô industrial, podem-se citar algumas definições internacionais de uso corrente:

“Máquina re-programável para realizar funções de manipulação, movimentação e usinagem em condições de interagir com o ambiente e dotada de certo grau de autonomia.” [IFTtoMM]

“Um dispositivo programável projetado para manipular e transportar peças, ferramentas ou implementos especializados de manufatura através de movimentos programáveis variados para a execução de tarefas específicas de manufatura.” [BARA]

“Dispositivo multifuncional, re-programável para realizar uma série de tarefas dedicadas à automação das atividades de um ambiente CIM.” [RIA]

O robô, como um sistema, consiste de elementos mecânicos, de acionamento e de controle, que são caracterizados e integrados da forma como segue:

Manipulador

É o corpo principal do robô e consiste dos elos, juntas e outros elementos estruturais. Dependendo do arranjo destes elos e juntas, o manipulador adquire uma determinada capacidade de movimentos, que resulta em uma área de trabalho específica. Os robôs podem, então, serem classificados pelo arranjo das juntas (que podem ser R, ou rotacionais, ou P, prismáticas) ou mesmo pela forma da área de trabalho resultante de seus manipuladores. São eles Cartesianos ou Retangulares (PPP), Cilíndricos (RPP), Esféricos (RRP) e Antropomorfos ou Articulados (RRR) e SCARA (RRP), conforme a Figura 2.1.

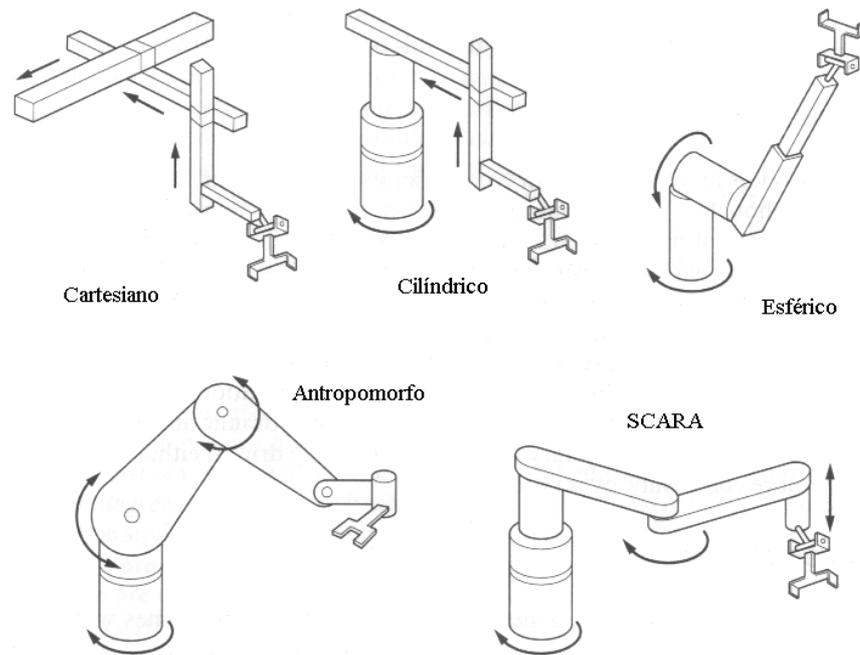


Figura 2.1 – Tipos de robôs.

Efetuator

É a parte conectada à última junta (pulso) do manipulador. Constitui a ferramenta que executa tarefas específicas, como manipulação, solda, pintura, etc. A possibilidade de serem acopladas ferramentas variadas à um mesmo robô confere a sua grande versatilidade. Pode ser comandado diretamente pelo controlador do robô ou por algum dispositivo de controle específico associado.

Atuadores

São os “músculos” do manipulador. Tipos comuns de atuadores são os servo-motores, motores de passo (quando se requer mais precisão), cilindros hidráulicos (mais capacidade de carga) ou cilindros pneumáticos (condição intermediária). São controlados pelo controlador do robô e são responsáveis pelos movimentos das juntas.

Sensores

São usados para coletar dados sobre o estado interno do robô ou para comunicar este com o ambiente externo. Sensores integrados enviam informações sobre cada junta ou elo ao controlador, que determina a configuração do robô. Algumas vezes, os robôs são equipados com sensores externos, como um sistema de visão, que o permite interagir com o ambiente e executar uma gama de tarefas que sem eles seria impossível, de modo automático.

Unidade de Controle

Constitui-se de um módulo principal, de controle geral do equipamento, constituído pela memória de controle e pelo processador principal, que recebe dados da memória de controle para

controlar os movimentos dos atuadores e coordenar esses movimentos com a informação dos sensores e de outro específico para algumas funções do robô, constituído pelo processador I/O e pelo processador de controle de eixos. O processador I/O coordena a entrada e saída de sinais oriundos de sensores ou de programas de processamento de dados externos. O processador de controle de eixos calcula os movimentos das juntas do robô, determina quanto e quão rápido uma junta deve se mover para alcançar desejada posição e velocidade e supervisiona as ações coordenadas do controlador principal e dos sensores.

Software

São empregados três níveis de softwares em um robô. Um é o sistema operacional do controlador. O segundo é o *software* robótico do processador, que calcula os movimentos necessários de cada junta baseados em equações cinemáticas do robô. O terceiro grupo é a soma de rotinas e programas de aplicação que são desenvolvidos para operar os dispositivos periféricos, como rotinas de visão, ou para executar tarefas específicas, podendo ser escrito em diversas linguagens de programação ou em linguagem própria a cada tipo de robô.

Nota-se, que os sistemas de visão integram um grupo externo ao robô e que a este pode ser adicionado, para expandir sua gama de tarefas possíveis.

2.2 Aplicações dos Robôs

Os robôs são equipamentos ideais para trabalhar em ambientes caracterizados por tarefas repetitivas, pesadas ou de risco para humanos ou ainda, que exigem precisão e repetitividade. São usados por muitas indústrias e nas mais variadas tarefas, oferecendo melhor performance e a custos mais baixos do que com operadores humanos. Apesar disso, algumas tarefas ficam dificultadas ou impossibilitadas quando se utiliza o robô “simples”, sem equipamentos periféricos, como um sistema de visão. A aplicação de sistemas de visão, associados à atuação do manipulador amplia ou possibilita melhor desempenho nas mais diversas tarefas, dentre elas podem-se citar as seguintes:

Carregamento

Onde robôs fornecem componentes para (ou os removem de) outras máquinas. O robô não executa trabalho no componente, apenas o transporta. Quando os componentes a serem transportados não estão sempre na mesma posição em relação ao robô, ou seja, quando o ponto de captura não é constante, a tarefa fica inviável de ser executada através de um programa que fornece coordenadas fixas ao robô. Neste caso, um sistema de visão pode fornecer estas coordenadas variáveis, pois permite ao robô “ver” onde a peça a ser capturada se encontra.

Pick and place

Operação em que o robô captura uma peça e a insere em um local específico, como em paletização, inserção de componentes, montagem simples ou rotinas similares. Um sistema de visão pode ser usado na mesma situação encontrada no carregamento.

Inspeção

De componentes, chassis, placas de circuito e outros produtos. Em geral, algum dispositivo específico está integrado ao sistema, como um sistema de visão, um dispositivo de raio-X, um detector ultra-sônico ou outro dispositivo similar. Muito comum é o uso de sistemas de visão em placas de circuito, onde seu diagrama é armazenado em um arquivo. As imagens das placas na linha de produção são comparadas com o diagrama correto armazenado em um computador e então aceitas ou rejeitadas, de acordo com os princípios de conformidade.

Seleção

Muito usada na agricultura, em controle de qualidade de alimentos, podendo ser um misto das tarefas de *pick and place* e inspeção. Imagens de frutas, por exemplo, podem ser analisadas com base na tonalidade das cores superficiais e então aceitas ou rejeitadas, sendo então capturadas e encaixotadas ou então descartadas [Blasco et al, 2003].

Aplicações médicas

Nesta área de aplicação cada vez mais cresce o interesse, e avançam as pesquisas, no campo das cirurgias à distância. Um manipulador robótico funciona como o braço do cirurgião, que controla o manipulador à distância e supervisiona o serviço através de um sistema de visão [Camarillo et al, 2004].

Operações em Ambientes perigosos

Situações ideais para aplicações da robótica. Para humanos trabalharem neste tipo de ambiente há a necessidade do uso de trajes especiais, sistemas de proteção e de suporte de vida, etc. Um robô dispensa estes cuidados, resguardando a saúde e integridade do operador, que pode supervisionar o serviço do robô, em um ambiente seguro, através de um sistema de visão [Colon et al, 2002].

Operações espaciais, submarinas e em ambientes remotos

Também podem ser explorados com mais facilidade por robôs. É extensivo o uso de robôs submarinos nas tarefas de pesquisa do leito oceânico e resgate de tripulações de submarinos [Maia, 2004]. Na área espacial, são enviados robôs exploradores há vários anos para a lua e outros corpos celestes além do uso de manipuladores robóticos, como o CANADARM da frota de ônibus espaciais americanos e o CANADARM-2 da Estação Espacial Internacional

[Sallaberger, 1997]. Em todos estes casos, há o emprego de sistemas de visão, quer para supervisão de trabalhos, posicionamento dos robôs ou para controle de trajetória automático, como nos *rovers* enviados à lua e marte [Tunstel e Howard, 2003].

Estas, e outras, tarefas de sistemas robotizados demonstram a gama de aplicações que os sistemas de visão encontram na robótica, pois, como em humanos, a capacidade da visão dota os robôs de um sofisticado mecanismo sensor que permite à máquina responder ao ambiente de uma maneira inteligente e flexível. O uso da visão e outros dispositivos sensores é motivado pela contínua necessidade de aumentar a flexibilidade e área de aplicação de sistemas robóticos. Enquanto sensores de proximidade, toque e força possuem um papel significativo na melhora da performance de um robô, a visão é reconhecida como a mais poderosa capacidade sensorial do robô.

2.3 Visão Robótica

Um grande campo de trabalho associado com sistemas de visão, processamento de imagem e reconhecimento de padrões leva à vários e diferentes tópicos de *hardware* e *software*. Esta informação está sendo acumulada desde os anos 50, com o crescente interesse no assunto de diferentes setores da indústria e economia. O grande número de trabalhos publicados a cada ano indica que existem muitas técnicas úteis surgindo constantemente na literatura. Ao mesmo tempo, isto também significa que várias destas técnicas devem estar disponíveis para serem usadas em outras aplicações.

Visão robótica pode ser definida como o processo de extrair, caracterizar e interpretar informações de imagens de um mundo tridimensional. Este processo, também comumente referido como *visão de máquina ou computador*, pode ser subdividido em seis áreas principais: (1) sensoreamento, (2) pré-processamento, (3) segmentação, (4) descrição, (5) reconhecimento e (6) interpretação [Fu et al, 1987]. Sensoreamento é o processo de captar a imagem. Pré-processamento trata de técnicas como redução de ruído e melhoramento de detalhes. Segmentação é o processo que divide uma imagem em objetos de interesse. Descrição lida com a computação de detalhes (tamanho, forma) conveniente para diferenciar um tipo de objeto de outro. Reconhecimento é o processo que identifica esses objetos (chaves, parafusos, blocos de motor, por exemplo). A interpretação consiste num conjunto de técnicas para fornecer um significado para os objetos reconhecidos.

É conveniente agrupar estas áreas de acordo com a sofisticação envolvida em sua implementação. Considera-se três níveis de processamento: visão de baixo, médio e alto nível.

Apesar de não haver limites definidos entre essas subdivisões, elas provêm uma útil ferramenta para categorizar os vários processos que são componentes inerentes de um sistema de visão de máquina. Associam-se esses processos com a visão de baixo nível, pois são elementares, no sentido de que eles podem ser considerados “reações automáticas”, não requerendo inteligência por parte do sistema de visão. Assim, deve-se tratar sensoramento e pré-processamento como funções de visão de baixo nível. Isto implica em passar do processo de formação da imagem propriamente a compensações como redução de ruído e finalmente a de traços primitivos da imagem como a descontinuidades de intensidade.

Associa-se com visão de médio nível os processos de extração, caracterização e marcação de componentes em uma imagem resultante de uma visão de baixo nível. Em termos das seis subdivisões, trata-se a segmentação, descrição e reconhecimento de objetos individuais como funções de médio nível. Visão de alto nível se refere a processos que visam simular raciocínio. Enquanto algoritmos de visão de baixo e médio nível encontram um razoável e bem-definido espectro de atividades, o conhecimento e entendimento de processos de visão de alto nível são consideravelmente mais vagos e especulativos.

Essas categorias e subdivisões sugerem como os sistemas de visão são geralmente implementados. Isto não implica que estas subdivisões representem um modelo da visão humana nem que sejam executados independentemente. Sabe-se, por exemplo, que reconhecimento e interpretação são funções inter-relacionadas nos seres humanos. Estes relacionamentos, entretanto, ainda não são entendidos ao ponto de poderem ser modelados analiticamente. A subdivisão de funções feita nesta discussão pode ser visualizada como uma aproximação prática para implementar o estado da arte em sistemas de visão robótica, dado o nível de conhecimento e as ferramentas analíticas correntemente disponíveis neste campo [Niku, 2001].

2.3.1 Imagem

Uma imagem é uma representação de uma cena real, tanto em preto-e-branco quanto a cores, e tanto em forma impressa quanto digital. Imagens impressas podem ser reproduzidas tanto à cores quanto em escalas de cinza ou por uma fonte simples de tinta. Por exemplo, para reproduzir uma fotografia com meios-tons, deve-se usar tintas múltiplas cinzas, as quais, quando combinadas, produzem uma imagem que é realista. Entretanto, em muitas aplicações de impressão, apenas uma cor de tinta é disponível (como tinta preta em papel branco em um jornal ou copiadora). Neste caso, todos os níveis de cinza podem ser produzidos mudando a razão de preto contra áreas brancas (o tamanho do ponto preto). Considere-se que um desenho a ser

impresso seja dividido em pequenas seções. Em cada seção, se a respectiva porção de tinta é menor quando comparada à área branca, a seção parecerá um cinza mais claro. Se a área de tinta preta for maior comparada à área branca, irá parecer um cinza mais escuro. Mudando o tamanho do ponto impresso, muitos níveis de cinza podem ser produzidos e coletivamente uma figura em escala de cinzas será impressa.

Ao contrário das imagens impressas, imagens de televisão e digitais são divididas em pequenas seções chamadas células de imagem, ou *pixels* (em imagens tridimensionais, eles são chamados células de volume ou *voxels*), onde o tamanho de todos os *pixels* é o mesmo, enquanto a intensidade de luz em cada *pixel* é variada para criar imagens cinzas. Lidando com imagens digitais, refere-se sempre a *pixels* de mesmo tamanho com intensidades variáveis.

Apesar de todas as cenas reais serem tridimensionais, imagens podem tanto ser bi ou tridimensionais. Imagens bidimensionais são usadas quando a profundidade da cena ou sua característica não precisam ser determinadas, como na definição do contorno da silhueta de um objeto. Neste caso, não é necessário determinar a profundidade de qualquer ponto do objeto. Outro exemplo é o uso de um sistema de visão para inspeção de uma placa de circuito impresso, onde não há necessidade de conhecer a relação de profundidade entre partes diferentes e desde que todas as partes são fixas a um plano, nenhuma informação sobre a superfície é necessária. Neste caso, a análise de uma imagem bidimensional e inspeção é suficiente.

Processamento de imagens tridimensionais lida com operações que requerem detecção de movimento, medida de profundidade, sensoriamento remoto, posicionamento relativo e navegação. Operações relacionadas com CAD/CAM também requerem processamento de imagens tridimensionais, como fazem muitas ferramentas de inspeção e reconhecimento de objetos [González-Galván et al, 2003]. Outras técnicas, como tomografia computadorizada, são também tridimensionais. Nesta técnica, tanto raios-X quanto pulsos ultra-sônicos são usados para obter imagens de uma fatia de um objeto em um instante para, posteriormente, todas as imagens serem postas juntas para criar uma imagem tridimensional das características internas do objeto.

Nas etapas iniciais da operação de um sistema de visão, este deve desempenhar basicamente duas funções: o processamento e a análise. Estes termos geram certa confusão, mas são duas tarefas distintas. *Processamento da imagem* relaciona a preparação de uma imagem para posterior análise e uso. Imagens capturadas por uma câmera ou outra técnica similar (como por scanner) não estão necessariamente na forma em que podem ser usadas por rotinas de análise de imagens. Algumas podem requerer melhoramento para redução de ruído, outras podem exigir simplificação e outras ainda podem necessitar realce, alteração, segmentação, filtragem, etc. *Processamento da imagem* é a soma de rotinas e técnicas que melhoram, simplificam, realçam

ou alteram uma imagem. *Análise da imagem* é a soma de processos nos quais uma imagem que foi preparada por processamento de imagem é analisada para extrair informação sobre a imagem para identificar objetos ou fatos sobre o objeto ou seu ambiente.

2.4 Captação da Imagem

A captação de imagens é realizada através de câmeras diretamente associadas com sistemas de visão. Elas podem ser de dois tipos: analógicas e digitais. Câmeras analógicas não são mais muito comuns, mas ainda são aplicadas; elas eram padrão em estações de televisão. Uma câmera filmadora de vídeo caseira é uma câmera com uma seção de gravação adicionada. Por outro lado, o mecanismo da aquisição de imagens é o mesmo como nas outras câmeras que não gravam imagens. Enquanto a imagem capturada é analógica ou digital, em sistemas de visão a imagem é eventualmente digitalizada. Na forma digital, todos os dados são binários e são armazenados em um arquivo de computador ou *chip* de memória.

2.4.1 Câmera Vidicon

Uma câmera vidicon é uma câmera analógica (transforma a imagem em um sinal analógico), cujo sinal, uma tensão elétrica variável (ou corrente) versus tempo, pode ser armazenado, digitalizado, transmitido ou reconstruído em uma imagem. A Figura 2.2 mostra um esquema simples de uma câmera vidicon. Com o uso de lentes, a cena é projetada em uma tela feita de duas camadas: um filme metálico transparente e um mosaico fotocondutor sensível à luz. O mosaico reage à intensidade variável da luz variando sua resistência. Como resultado, conforme a imagem é projetada nele, a magnitude da resistência em cada local varia com a intensidade da luz. Um canhão de elétrons gera e envia um raio catódico contínuo (um feixe de elétrons com carga negativa) através de dois pares de capacitores (defletores) que são perpendiculares entre si. Dependendo da carga de cada par de capacitores, o raio de elétrons é defletido para cima ou para baixo e para esquerda ou direita e é projetado no mosaico fotocondutor. A cada instante, assim que o feixe de elétrons atinge o mosaico, a carga é conduzida ao filme metálico e pode ser medida na porta de saída como diferença de potencial.

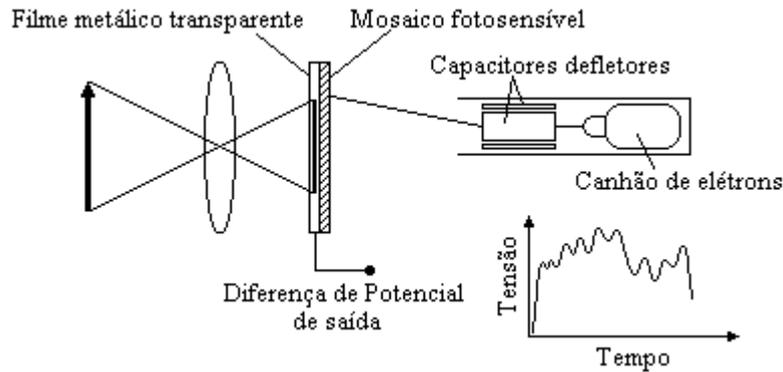


Figura 2.2 – Esquema de uma câmera Vidicon.

Considerando-se que rotineiramente alteram-se as cargas nos dois capacitores e o raio seja defletido para ambos os lados, para cima e para baixo, ocorrerá uma varredura do mosaico. Como o raio varre a imagem, em cada instante a saída é proporcional à resistência do mosaico ou proporcional à intensidade da luz no mosaico. Lendo a voltagem de saída continuamente, uma representação analógica da imagem pode ser obtida.

Para criar imagens em movimento em televisões, usando como exemplo o sistema NTS (*National Television Standards*), a imagem é lida e reconstruída 30 vezes por segundo. Para os olhos humanos, que possuem uma histerese temporária de cerca de 1/10 de segundo, imagens mudando 30 vezes por segundo são percebidas como contínuas e estando em movimento. A imagem é dividida em duas sub-imagens de 240 linhas, interligadas entre si. Logo, uma imagem de televisão é composta de 480 linhas de imagens, mudando 30 vezes por segundo. Para retornar o raio para o topo do mosaico, outras 45 linhas são usadas, criando um total de 525 linhas. Em alguns sistemas, o padrão pode ser de 625 linhas. A Figura 2.3 mostra uma varredura em uma câmera vidicon.

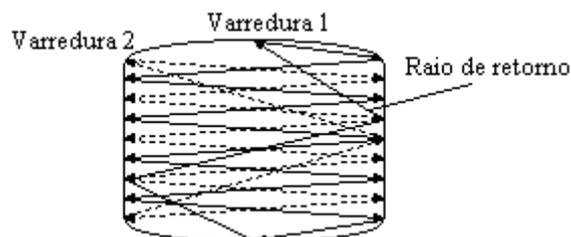


Figura 2.3 – Varredura da câmera Vidicon.

Para o sinal ser transmitido, este é usualmente modulado em frequência (FM); isto é, a frequência do sinal portador é uma função da amplitude do sinal. O sinal é transmitido e é recebido por um receptor, onde é modulado na forma do sinal original, criando uma tensão variável em relação ao tempo. Para recriar a imagem – por exemplo, em um aparelho de

televisão – esta tensão deve ser convertida em uma imagem. Para isto, a voltagem é inserida em um tubo de raios catódicos (CRT) com um canhão de elétrons e capacitores de deflexão semelhantes, como em uma câmera Vidicon. A intensidade do raio de elétrons na televisão é então proporcional à voltagem do sinal e varrida similarmente como na câmera. Na televisão, entretanto, o raio é projetado em um material à base de fósforo na tela, que fosforesce proporcionalmente à intensidade do raio, recriando a imagem.

Para imagens coloridas, a imagem projetada é decomposta em três cores, vermelho, verde e azul (RGB). O mesmo processo é repetido para as três imagens e três sinais simultâneos são produzidos e transmitidos. No aparelho de televisão, três canhões de elétrons regeneram três imagens simultâneas em RGB na tela, exceto quando a tela possui três conjuntos de pequenos pontos (*pixels*) que reagem fosforescendo em cores RGB e são repetidos sobre a tela inteira. Um sinal tanto pode ser transmitido, gravado para uso posterior, digitalizado ou inserido em um monitor para visualização direta.

2.4.2 Câmera Digital

A câmera digital é baseada na tecnologia do estado sólido. Como em outras câmeras, um conjunto de lentes é usado para projetar a área de interesse sobre a área da imagem de interesse da câmera. A maior parte da câmera é uma área de imagem de silício em estado sólido que possui centenas de milhares de extremamente pequenas áreas fotossensíveis chamadas *photosites* sobre ela. Cada pequena área é um *pixel*. Assim que a imagem é projetada sobre a área da imagem, em cada *pixel* uma carga é criada proporcionalmente à intensidade da luz em cada ponto (uma câmera digital é também chamada de dispositivo de carga acoplada ou câmera CCD e dispositivo de carga integrada ou câmera CID). O conjunto de cargas, lido seqüencialmente, pode ser uma representação dos *pixels* da imagem [Okigawa, 1992].

Podem haver até 520.000 *pixels* em uma área de fração de polegada ($3/16 \times 1/4$), o que torna impossível haver conexões diretas por cabo para todos estes *pixels* para medir a carga em cada um. Para ler este grande número de *pixels*, 30 vezes por segundo as cargas são movidas para registradores opticamente isolados próximos a cada *photosite*. Posteriormente, são movidas para uma linha de saída e então lidas [Wilson, 1984]. O resultado é que a cada 30 avos de segundo as cargas em cada *pixel* são lidas seqüencialmente e gravadas. A saída é uma representação discreta de uma imagem – uma tensão capturada no tempo – como mostrado na Figura 2.4.

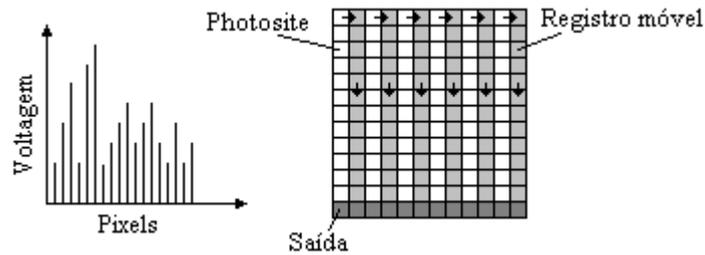


Figura 2.4 – Modelo de captura de dados de imagem.

Similar às câmeras CCD para luzes visíveis, câmeras de longo comprimento de onda para infravermelho permitem uma imagem similar à de televisão das emissões infravermelhas de uma cena [NASA, 1999].

As tensões tomadas no processo descrito são digitalizadas através de um conversor analógico-digital e então armazenadas no computador em um formato de imagem como TIFF, JPEG, BMP, etc., ou mostradas em um monitor. Sendo digitalizada, a informação guardada é uma coleção de 0's e 1's que representa a intensidade da luz em cada *pixel*; uma imagem digitalizada não é nada mais que um arquivo de computador que contém essa coleção de 0's e 1's, seqüencialmente guardados para representar a intensidade da luz em cada *pixel*. Os arquivos podem ser acessados e lidos por um programa computacional, serem duplicados e manipulados, ou ainda serem reescritos em um formato diferente. Rotinas de visão geralmente acessam estas informações, executam alguma função nos dados e tanto mostram o resultado quanto gravam o resultado em um novo arquivo.

Uma imagem com diferentes níveis de cinza em cada *pixel*, é chamada de uma imagem cinza. Os valores de cinza são digitalizados, gerando uma imagem de 0's e 1's (representação binária) que são subseqüentemente guardados ou mostrados. Uma imagem a cores é obtida superpondo três imagens de tons de vermelho, verde e azul, cada uma variando de intensidade e equivalente a uma imagem cinza. Quando a imagem é digitalizada, ela também terá valores de 0's e 1's para cada tom. Uma imagem binária é uma imagem que em cada *pixel* pode ser totalmente claro ou totalmente escuro, representado por 0 ou 1. Para obter-se uma imagem binária, em muitos casos uma imagem cinza é convertida usando o histograma da imagem e um valor chamado limite. Um histograma determina a distribuição dos diferentes níveis de cinza. Pode-se tomar um valor que melhor determina um nível com menor distorção e usar este valor como limite para designar 0's (ou “desligado”) à todos os *pixels* cujos níveis de cinza estão abaixo deste valor ou designar 1's (ou “ligado”) para todos os *pixels* cujos valores de cinza estão acima deste limite. Mudando-se o limite, muda-se a imagem binária. A vantagem de uma

imagem binária é que ela requer muito menos memória e pode ser processada muito mais rapidamente do que imagens cinzas ou coloridas [Shi, 1997].

2.5 Técnicas de Iluminação

Iluminação de uma cena é um fator importante que freqüentemente afeta a complexidade de algoritmos de visão. Iluminação arbitrária do ambiente freqüentemente é inaceitável porque pode resultar em imagens com pouco contraste, reflexões, sombras e detalhes estranhos. Um sistema de iluminação bem projetado trata a cena de tal modo que a complexidade da imagem adquirida é minimizada, enquanto a informação requerida para a detecção de objetos e sua extração é melhorada.

A Figura 2.5 mostra cinco dos principais esquemas usados para iluminar um espaço de trabalho com robôs [Mundy, 1977]. O método mais simples é o da iluminação direta, onde uma fonte de luz é disposta diretamente acima do objeto de interesse, como mostrado na Figura 2.5(a).

A iluminação difusa, mostrada na Figura 2.5(b), pode ser empregada para objetos lisos com superfícies regulares. Este esquema é geralmente empregado em aplicações onde características da superfície são importantes. Iluminação traseira, como mostrado na Figura 2.5(c), produz uma imagem preto-e-branco (binária). Esta técnica é ideal para aplicações onde silhuetas de objetos são suficientes para reconhecimento ou outras medidas.

A iluminação estruturada, mostrada na Figura 2.5(d), consiste de projetar pontos, listras ou grades na superfície de trabalho. Esta técnica tem duas vantagens importantes: primeiro, ela estabelece um padrão de luz conhecido na superfície de trabalho e distúrbios neste padrão indicam a presença de um objeto, simplificando o problema de detecção de objetos; segundo, analisando o modo como o padrão é distorcido é possível visualizar melhor as características tridimensionais do objeto.

A iluminação direcional, mostrada na Figura 2.5(e), é útil principalmente para inspeção superficial de objetos. Defeitos na superfície, como reentrâncias e riscos, podem ser detectados usando um feixe de luz direcionado (como um feixe de laser) e medindo o espalhamento de luz. Em superfícies livres de defeitos há pouco espalhamento. Por outro lado, a presença de defeitos aumenta a quantidade de luz espalhada, facilitando sua detecção.

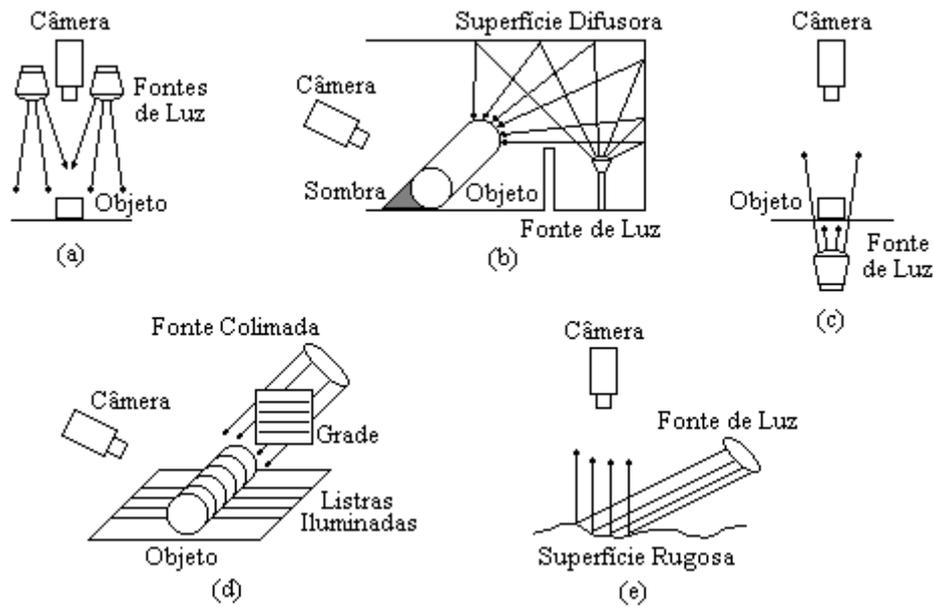


Figura 2.5 – Formas de iluminação

2.6 Pré-Processamento de Imagens

Existem diversas técnicas de pré-processamento de imagens passíveis de serem empregadas em visão robótica, mas poucas cumprem os requisitos de velocidade computacional e baixo custo de implementação, que são elementos essenciais de um sistema de visão industrial [Fu et alli, 1987]. Os métodos básicos podem ser resumidos a dois: os baseados em técnicas de domínio-freqüência e os baseados em técnicas de domínio-espço.

No processo de domínio-freqüência, o espectro de freqüência da imagem é usado para alterar, analisar ou processar a imagem. Neste caso, *pixels* individuais e seus conteúdos não são usados, sendo uma representação em freqüência da imagem inteira usada no processo. No processo de domínio-espço, o processo é aplicado à *pixels* individuais na imagem. Como resultado, cada *pixel* é afetado diretamente pelo processo. As duas técnicas são igualmente importantes e poderosas e são usadas para diferentes propósitos, ressaltando-se que apesar de serem usadas diferentemente, elas são relacionadas.

2.6.1 Domínio-Freqüência

Domínio-freqüência se refere a um conjunto de *pixels* resultantes da obtenção da transformada de Fourier de uma imagem. O conceito de freqüência é normalmente usado na interpretação da transformada de Fourier e vem do fato de que esta transformada em particular é composta de senóides complexas. Devido a seus extensivos requisitos de processamento,

métodos de domínio-freqüência não são tão usados em visão robótica como as técnicas de domínio-espço. Entretanto, a transformada de Fourier tem um importante papel em áreas como a análise de movimento de objetos e descrição de objetos. Em adiço, muitas tcnicas espaciais para melhoramento e restauraço so fundadas em conceitos cujas origens podem ser traçadas para a formulaço da transformada de Fourier [Gonzales e Wintz, 1977].

Considerando-se funçes discretas de uma varivel, $f(x)$, para $x = 0, 1, 2, \dots, N-1$. A transformada de Fourier $F(u)$, para $u = 0, 1, 2, \dots, N-1$, em relaço a $f(x)$ é dada por:

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) e^{-j2\pi ux / N} \quad (1)$$

Nesta equaço, $j = \sqrt{-1}$ e u é a chamada varivel freqüência. A transformada inversa de Fourier para $F(u)$ retorna a funço $f(x)$, para $x = 0, 1, 2, \dots, N-1$, sendo definida por:

$$f(x) = \sum_{u=0}^{N-1} F(u) e^{j2\pi ux / N} \quad (2)$$

A validade destas expresses, chamadas de *Par Transformada de Fourier*, é facilmente verificada substituindo a Equaço (1) para $F(u)$ na Equaço (2), ou vice-versa. Em ambos os casos obtm-se uma identidade.

Uma implementaço direta da Equaço (1) para $u = 0, 1, 2, \dots, N-1$ pode requerer uma ordem de N^2 adiçes e multiplicaçes. O uso de uma Transformada Rpida de Fourier (FFT) reduz significativamente este nmero para $N \log_2 N$, onde N é assumido como uma potncia inteira de 2. Comentrios similares aplicam-se à Equaço (2) para $x = 0, 1, 2, \dots, N-1$. Grande nmero de algoritmos FFT est disponvel em uma variedade de linguagens computacionais [Fu et alli, 1987].

O par bi-dimensional da transformada de Fourier de uma imagem $N \times N$, para $u, v = 0, 1, 2, \dots, N-1$, é definido como:

$$F(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(u x + v y) / N} \quad (3)$$

E para $x, y = 0, 1, 2, \dots, N-1$:

$$f(x, y) = \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u, v) e^{-j2\pi(ux+vy)/N} \quad (4)$$

É possível mostrar através de alguma manipulação que cada uma destas equações pode ser expressa como somatórios unidimensionais separados na forma mostrada na Equação (1). Isto leva a um procedimento para cálculo da transformada de Fourier bi-dimensional usando apenas um algoritmo FFT unidimensional: primeiro calcula-se a transformada de cada linha de $f(x,y)$, produzindo uma disposição bi-dimensional de resultados intermediários. Estes resultados são multiplicados por N e a transformada unidimensional de cada coluna é calculada. O resultado final é $F(u,v)$. O mesmo se aplica para calcular $f(x,y)$ dado $F(u,v)$. A ordem do cálculo de uma aproximação linha-coluna pode ser invertida para um formato coluna-linha sem afetar o resultado final.

A transformada de Fourier pode ser usada em vários caminhos de um sistema de visão. Por exemplo, tratando a fronteira de um objeto como uma disposição unidimensional de pontos e calculando sua transformada de Fourier, valores selecionados de $F(u)$ podem ser usados para descrever a forma do contorno. A transformada unidimensional de Fourier também é usada como uma poderosa ferramenta para detecção de movimento de objetos. Aplicações da transformada discreta de Fourier bidimensional na reconstrução, melhoramento e restauração da imagem também são abundantes e a utilidade deste método na visão robótica industrial está um tanto restrita devido a seus extensivos requisitos computacionais necessários para implementar a transformada. Cita-se também que a transformada contínua bidimensional de Fourier pode ser calculada por meios ópticos. Este método, que requer o uso de instrumentos ópticos precisamente alinhados, pode ser usado em ambientes industriais para tarefas como inspeção de superfícies acabadas de metal [Goodman, 1968].

2.6.2 Domínio-Espaço

Domínio-espaco refere-se ao agregado de *pixels* que compõem uma imagem e métodos domínio-espaco são procedimentos que operam diretamente nesses *pixels*. Funções de pré-processamento no domínio-espaco podem ser expressas como:

$$g(x, y) = h[f(x, y)] \quad (5)$$

Onde $f(x,y)$ é a imagem de entrada, $g(x,y)$ é a imagem resultante (pré-processada) e h é um operador em f , definido sobre a vizinhança de (x,y) . Também é possível deixar h operar em um conjunto de imagens de entrada, como uma soma *pixel por pixel* de K imagens para redução de ruído.

O principal método usado para definir uma vizinhança de (x,y) é usar uma sub-imagem quadrada ou retangular centrada em (x,y) , como mostrado na Figura 2.6. O centro da sub-imagem é movido *pixel por pixel* começando pelo canto superior esquerdo e aplicando o operador em cada localização (x,y) para tornar $g(x,y)$. Ainda que outras formas de vizinhança, como um círculo, sejam algumas vezes utilizadas, disposições quadradas são as predominantes devido à sua fácil implementação.

A forma mais simples de h ocorre quando a vizinhança é 1×1 e, logo, g depende apenas do valor de f em (x,y) . Neste caso h se torna uma transformação da forma:

$$s = T(r) \quad (6)$$

Onde, por simplicidade, usa-se s e r como variáveis denotando, respectivamente, a intensidade de $f(x,y)$ e $g(x,y)$ em qualquer ponto (x,y) .

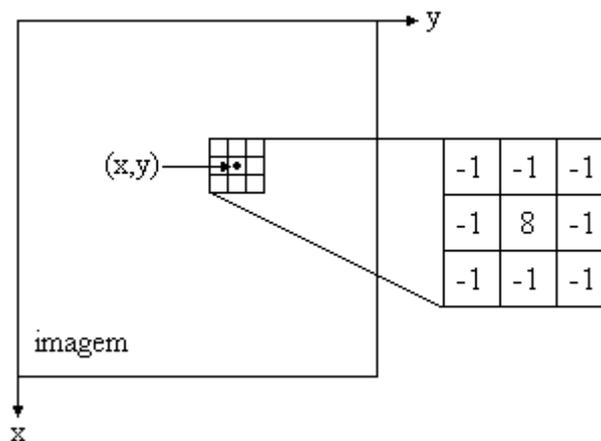


Figura 2.6 – Vizinhança 3×3 de um ponto da imagem e detalhe da máscara de convolução.

Uma das técnicas domínio-espaço usadas mais frequentemente é baseada no uso das chamadas máscaras de convolução (também chamadas de *templates*, janelas ou filtros). Basicamente, uma máscara é uma pequena disposição bi-dimensional (3×3), como a mostrada na Figura 2.6, cujos coeficientes são escolhidos para detectar uma dada propriedade numa imagem. Como introdução a este conceito, supõe-se que tem-se uma imagem de intensidade constante que contém *pixels* isolados cujas intensidades são diferentes do fundo. Estes pontos

podem ser detectados usando a máscara mostrada na Figura 2.6. O procedimento é como segue: o centro da máscara (chamado 8) é movido em torno da imagem, como indicado acima. Em cada posição de *pixel* na imagem, multiplica-se cada *pixel* contido na área da máscara pelo correspondente coeficiente da máscara; isto é, o *pixel* no centro da máscara é multiplicado por 8, enquanto os 8 vizinhos são multiplicados por -1 . Os resultados destas nove multiplicações são então somados. Se todos os *pixels* sob a área da máscara têm o mesmo valor (fundo constante), a soma será zero. Se, por outro lado, o centro da máscara estiver localizado em um dos pontos isolados, a soma será diferente de zero. Se o ponto isolado é uma posição fora do centro, a soma será também diferente de zero, mas a magnitude da resposta será mais fraca. Estas respostas mais fracas podem ser eliminadas comparando a soma contra a extremidade.

Como mostrado na Figura 2.7, se w_1, w_2, \dots, w_9 representam coeficientes da máscara e considerando os 8 vizinhos de (x,y) , pode-se generalizar o exposto executando a seguinte operação:

$$h[f(x, y)] = w_1 f(x-1, y-1) + w_2 f(x-1, y) + w_3 f(x-1, y+1) + w_4 f(x, y-1) + w_5 f(x, y) + w_6 f(x, y+1) + w_7 f(x+1, y-1) + w_8 f(x+1, y) + w_9 f(x+1, y+1) \quad (7)$$

Em uma vizinhança 3×3 de (x,y) .

w_1 $(x-1, y-1)$	w_2 $(x-1, y)$	w_3 $(x-1, y+1)$
w_4 $(x, y-1)$	w_5 (x, y)	w_6 $(x, y+1)$
w_7 $(x+1, y-1)$	w_8 $(x+1, y)$	w_9 $(x+1, y+1)$

Figura 2.7 – Máscara 3×3 mostrando coeficientes e coordenadas de *pixel*.

2.6.3 Conectividade

A conectividade estabelece quando *pixels* estão conectados, ou seja, quando possuem as mesmas propriedades, como pertencer à mesma região ou ao mesmo objeto, ter uma textura similar, etc. Para estabelecer a conectividade de *pixels* vizinhos, primeiro deve-se escolher um critério de conectividade. Destes, há três critérios fundamentais para o processamento de

imagens bidimensionais: conectividade+4 ou $\times 4$, conectividade H6 ou V6 e conectividade-de-8. Em três dimensões, a conectividade entre *voxels* (células de volume) é de 6 até 26. Os seguintes termos de relacionamento de um *pixel* p são definidos com respeito à Figura 2.8:

Conectividade +4: em relação aos quatro *pixels* imediatamente acima, abaixo, à esquerda e à direita (b, g, d e e).

Conectividade $\times 4$: em relação aos quatro *pixels* imediatamente diagonais aos quatro lados (a, c, f e h).

Conectividade H6: em relação aos seis *pixels* vizinhos nas duas linhas acima e abaixo (a, b, c, f, g e h).

Conectividade V6: em relação aos seis *pixels* vizinhos nas duas colunas à direita e à esquerda (a, d, f, c, e e h).

Conectividade-de-8: em relação a os oito *pixels* que o circundam (a, b, c, d, e, f, g e h).

a	b	c
d	p	e
f	g	h

Figura 2.8 – Vizinhança de *pixel* p .

Para um *pixel* $p(x,y)$ os *pixels* relevantes são como segue:

Conectividade +4: $(x+1, y), (x-1, y), (x, y+1), (x, y-1)$

Conectividade $\times 4$: $(x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)$

Conectividade H6: $(x-1, y+1), (x, y+1), (x+1, y+1), (x-1, y-1), (x-1, y), (x+1, y-1)$

Conectividade V6: $(x-1, y+1), (x-1, y), (x-1, y-1), (x+1, y+1), (x+1, y), (x+1, y-1)$

Conectividade-de-8: $(x-1, y-1), (x, y-1), (x+1, y-1), (x-1, y), (x+1, y), (x-1, y+1), (x, y+1), (x+1, y+1)$

2.7 Processamento de Imagens

As técnicas de processamento da imagem são usadas para realçar, melhorar ou alterar uma imagem e prepará-la para sua análise. Usualmente, durante o processamento da imagem não se extrai informação da mesma. A intenção é remover erros, informações triviais ou informações que até podem ser importantes, mas inúteis para os resultados que se desejam, e melhorar a

imagem. Uma imagem obtida com o objeto se movendo pode resultar em uma imagem que não é clara. Para sua utilização é desejável reduzir ou remover o borrão na imagem antes que informações sobre o objeto (como sua natureza, forma, localização, orientação, etc.) possam ser determinadas.

O processamento da imagem é dividido em muitos sub-processos, incluindo análise de histograma, limiarização, detecção de borda, segmentação, modelamento de região, entre outros.

2.7.1 Histograma da Imagem

Um histograma é uma representação do número total de *pixels* de uma imagem em cada nível de cinza. A informação do histograma é usada em vários processos, incluindo limiarização. A informação de um histograma pode ajudar na determinação do ponto de interrupção quando a imagem é transformada em valores binários. Também pode ser usada para decidir onde há níveis de cinza predominantes na imagem. O histograma de uma imagem digital com níveis de cinza no intervalo $[0, L-1]$ é uma função discreta, do tipo:

$$p(r_k) = \frac{n_k}{n} \quad (8)$$

Onde, r_k é o k -ésimo nível de cinza, n_k é o número de *pixels* com esse nível de cinza, n é o número total de *pixels* na imagem, $k = 0, 1, 2, \dots, L-1$ e L é o número de níveis de cinza. Genericamente analisando, $p(r_k)$ dá uma estimativa da probabilidade de ocorrência do nível de cinza r_k . Um gráfico dessa função para todos os valores de k fornece uma descrição global da aparência de uma imagem.

A Figura 2.9 mostra os histogramas de quatro tipos básicos de imagens. O gráfico ilustrado em 2.9(a) mostra que os níveis de cinza estão concentrados em direção à extremidade escura do intervalo de níveis de cinza. Assim, este histograma corresponde a uma imagem com características predominantemente escuras. O oposto se verifica na Figura 2.9 (b). O histograma mostrado em 2.9(c) tem uma forma estreita que indica uma escala dinâmica pequena, portanto, corresponde a uma imagem de baixo contraste. Como todos os níveis de cinza ocorrem no meio da escala, a imagem pareceria um cinza denso. Finalmente, a Figura 2.9(d) mostra um histograma com espalhamento significativo, correspondendo a uma imagem de alto contraste.

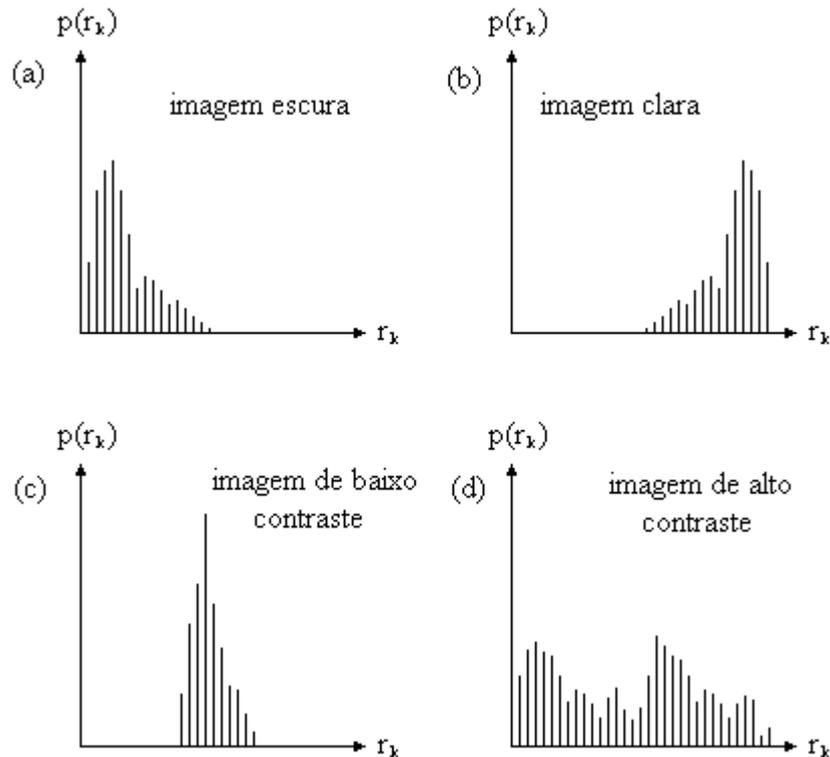


Figura 2.9 – Histogramas de 4 tipos básicos de imagens.

2.7.2 Limiarização

Limiarizar a imagem é uma das principais técnicas usadas em sistemas de visão industrial para detecção de objetos, especialmente em aplicações requerendo alta capacidade de processamento. É o processo de dividir uma imagem em diferentes porções (ou níveis) selecionando um certo nível de cinza como limite, comparando cada valor de *pixel* com o limite e então assinalando o *pixel* a diferentes porções ou níveis, dependendo se o nível de cinza está abaixo do limite (desligado, zero ou não pertence) ou acima do limite (ligado, 1 ou pertence). Limiarização pode ser executada tanto com um nível único quanto para vários níveis, nos quais a imagem processada é dividida em camadas, cada uma com um limite selecionado.

Rotinas mais comuns, usadas para imagens binárias, têm o objeto claro e o fundo totalmente escuro, ou o contrário. Esta condição pode ser alcançada em iluminação controlada em condições industriais, mas não é possível de ser alcançada em outros ambientes. Em imagens binárias, os *pixels* são “ligados” ou “desligados” e, assim, escolher um limite é simples. Em outras situações, a imagem terá múltiplos níveis de cinza e seu histograma exibe uma distribuição bi-modal. Neste caso, o vale é escolhido como o valor limite. Técnicas mais avançadas usam informações estatísticas e características de distribuição dos *pixels* da imagem para encontrar um valor limite. Como o valor limite varia, varia a imagem.

Portanto, pode-se ver a limiarização de imagens como uma operação que envolve testes contra uma função T da forma [Gonzalez e Woods, 2003]:

$$T = T[x, y, p(x, y), f(x, y)] \quad (9)$$

Onde $f(x,y)$ é a intensidade do ponto (x,y) e $p(x,y)$ denota alguma propriedade local deste ponto, por exemplo, a média da intensidade de uma vizinhança centrada em (x,y) . Uma imagem limiarizada $g(x,y)$ é definida como

$$g(x, y) = \begin{cases} 1 & \text{se } f(x, y) > T \\ 0 & \text{se } f(x, y) \leq T \end{cases} \quad (10)$$

Examinando $g(x,y)$, encontra-se *pixels* rotulados 1 (ou qualquer outro nível de cinza conveniente) corresponde à objetos, enquanto *pixels* 0 correspondem ao fundo. A Figura 2.10 exemplifica esta operação com um objeto genérico e diversas sombras ou elementos indesejados na cena captada. Após a limiarização, a cena fica “limpa”, podendo-se tratar apenas com o objeto.

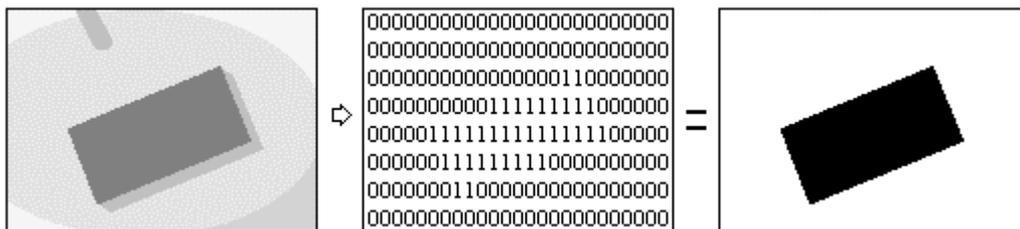


Figura 2.10 – Processo de limiarização.

Quando T depende apenas de $f(x,y)$, a limiarização é chamada global. Se T depende tanto de $f(x,y)$ quanto de $p(x,y)$, então a limiarização é chamada local. Se, além disso, T depende das coordenadas espaciais de x e y , é chamada de limitação dinâmica.

2.7.3 Detecção da Borda

A detecção da borda tem um papel importante na visão de máquina, servindo como pré-processamento inicial para numerosos algoritmos de detecção de objetos. Uma borda é o limite entre duas regiões com propriedades relativamente diferentes de níveis de cinza. Na detecção de borda, assume-se que essas regiões são homogêneas o suficiente para que a transição entre duas

regiões possa ser determinada com base apenas nas discontinuidades dos níveis de cinza. Caso isso não seja válido, as técnicas da limiarização e segmentação são mais aplicáveis [Gonzales e Woods, 2003].

Basicamente, a idéia por trás de muitas técnicas de detecção da borda é a computação de um operador diferencial local. Analisando-se a imagem de uma linha escura sobre um fundo claro, como mostrado na Figura 2.11, com o perfil de intensidade ao longo de uma linha de varredura horizontal da imagem e a primeira e segunda derivadas do perfil. Nota-se do perfil que uma borda (transição do escuro para o claro) é modelado como uma rampa, ao invés de uma abrupta mudança de intensidade. Isto representa o fato de que bordas em imagens digitais são, geralmente, levemente borradas como resultado da captação.

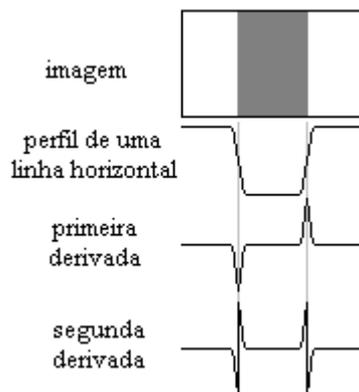


Figura 2.11 – Detecção da borda por operadores de derivação.

A primeira derivada é zero em todas as regiões de intensidade constante e assume um valor constante durante uma transição de intensidade. A segunda derivada é zero em todos os locais, exceto no início e fim de uma transição de intensidade. Baseado nisto, é evidente que a magnitude da primeira derivada pode ser usada para detectar a presença de uma borda, enquanto o sinal da segunda derivada pode ser usado para determinar quando um *pixel* da borda está no lado claro (fundo) ou escuro (objeto) da borda, respectivamente negativo e positivo.

São empregados em detecção da borda os operadores de gradiente, sendo que o gradiente de uma imagem $f(x,y)$ no ponto (x,y) é definido como o vetor

$$\nabla F = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (11)$$

Sabe-se da análise vetorial que o vetor gradiente aponta na direção de mais rápida taxa de mudança de f na posição (x,y) . Em detecção de bordas, a magnitude desse vetor é uma quantidade importante chamada simplesmente de gradiente e denotada por ∇f , onde:

$$\nabla f = \text{mag}(\nabla F) = [G_x^2 + G_y^2]^{1/2} \quad (12)$$

Essa quantidade equivale à maior taxa de aumento de $f(x,y)$ por unidade de distância na direção de ∇f . É prática comum aproximar o gradiente por valores absolutos:

$$\nabla f \approx |G_x| + |G_y| \quad (13)$$

Esta aproximação é consideravelmente mais fácil de implementar, particularmente quando hardware dedicado está sendo empregado.

Nota-se pela Equação (12) que o cálculo do gradiente é baseado em obter-se as derivadas de primeira ordem $\partial f / \partial x$ e $\partial f / \partial y$. Há vários modos de se implementar derivação de maneira digital, mas os operadores de Sobel têm a vantagem de fornecer efeitos de diferenciação e suavização. Sabendo que a derivação aumenta o ruído, a suavização é uma característica atrativa dos operadores de Sobel. Partindo-se da Figura 2.12 tem-se que as derivadas baseadas nas máscaras de Sobel são [Gonzales e Woods, 2003]:

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \quad (14)$$

$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \quad (15)$$

Onde z são os níveis de cinza dos *pixels* sobrepostos pelas máscaras em qualquer posição da imagem, sendo que estas são usadas para o cômputo de G_x e G_y no ponto central da região 3×3 . O valor do gradiente no ponto central da máscara é obtido utilizando-se a Equação (12) ou a (13). A máscara é deslocada para o próximo *pixel* e o procedimento é repetido. O resultado final é uma imagem de gradiente do mesmo tamanho da imagem original.

região 3 × 3 da imagem		
z ₁	z ₂	z ₃
z ₄	z ₅	z ₆
z ₇	z ₈	z ₉

máscara para cálculo de G _x		
-1	-2	-1
0	0	0
1	2	1

máscara para cálculo de G _y		
-1	0	1
-2	0	2
-1	0	1

Figura 2.12 – Máscaras de Sobel.

2.7.4 Segmentação Orientada a Regiões

O objetivo da segmentação é partir uma imagem em regiões. Em limitação busca-se definir regiões através das propriedades dos *pixels*, como suas intensidades, e em detecção da borda, isto é feito baseado na intensidade das descontinuidades. Em segmentação, buscam-se estas regiões diretamente.

Sendo R a região inteira da imagem, pode-se ver a segmentação como um processo de partir R em n subregiões, R_1, R_2, \dots, R_n , tal que [Gonzalez e Woods, 2003]:

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

Onde $P(R_i)$ é um predicado lógico definido sobre os pontos do conjunto R_i e \emptyset é o conjunto vazio.

A condição (1) indica que a segmentação deve ser completa, isto é, cada *pixel* deve estar em uma região. A condição (2) requer que pontos na região devam ser conectados. A condição (3) indica que as regiões não devem ter partes comuns. A condição (4) diz quais as propriedades que devem ser satisfeitas pelos *pixels* em uma região segmentada. A condição (5) indica que as regiões R_i e R_j são diferentes no sentido do predicado P .

A segmentação orientada a regiões pode ser implementada através de dois processos básicos: o crescimento de regiões através da agregação de *pixels* e a partição seguida de fusão de regiões.

Crescimento de Região

É um procedimento que agrupa *pixels* ou sub-regiões em regiões maiores. A mais simples abordagem é a agregação de *pixels*, onde se inicia com um conjunto de pontos “semente” e deles crescem regiões por adição dos vizinhos que tem propriedades similares (como textura, intensidade ou cor). A escolha de três sementes em uma imagem, por exemplo, resultaria em, no máximo, três regiões. Diferentes critérios empregados para a agregação de *pixels* e limiares arbitrados para esta operação forneceriam resultados finais diferentes. Note-se que após executada a segmentação, qualquer ponto pertencente a uma dada região que tivesse sido usado como semente levaria ao mesmo resultado [Fu et al, 1987].

Dois problemas imediatos da segmentação são a seleção de sementes iniciais que melhor representem as regiões de interesse e a seleção de propriedades para incluir pontos nas várias regiões durante o processo de crescimento. Selecionar um ou mais pontos iniciais pode se basear na natureza do problema. Por exemplo, em aplicações militares de imagem infravermelho, alvos de interesse são mais quentes (e logo parecem mais brilhantes) que o fundo. Escolher os *pixels* mais brilhantes é um ponto inicial para um algoritmo de crescimento de região. Quando uma informação anterior não é disponível, pode-se calcular em cada *pixel* o mesmo conjunto de propriedades que irão ser usadas para designar pontos a regiões durante o processo de crescimento. Se o resultado deste cálculo mostrar aglomerados de valores, então os *pixels* cujas propriedades colocam eles próximos do centróide destes aglomerados podem ser usados como sementes.

A seleção de critérios de similaridade é dependente não apenas do problema em consideração, mas também do tipo de imagem disponível. Onde necessita-se de imagens coloridas para análise, a disponibilidade apenas de imagens monocromáticas se tornaria um problema difícil de lidar. No entanto, a disponibilidade de imagens multiespectrais e com outros dados complementares é uma exceção, ao invés de regra, em sistemas de visão. Tipicamente, a análise da região deve ser feita usando um conjunto de descritores baseados em propriedades de intensidade e espaciais de uma simples fonte de imagem.

Partição e Fusão de Regiões

Uma alternativa ao crescimento é subdividir uma imagem em um conjunto de regiões arbitrárias e então fundi-las e/ou parti-las na tentativa de satisfazer as condições impostas. Um algoritmo de partição e fusão que iterativamente trabalha para satisfazer estas definições pode ser exposto como segue.

Seja R a região inteira da imagem e P um predicado. Assumindo uma imagem quadrada, um método para segmentar R é sucessivamente subdividir em regiões quadradas cada vez menores de maneira que, para qualquer região R_i , $P(R_i) = \text{VERDADEIRO}$. Se $P(R) = \text{FALSO}$, divide-se a imagem em quadrantes. Se P é falso para qualquer quadrante, subdivide-se em sub-quadrantes, e assim sucessivamente [Fu et alli, 1987].

Usando-se apenas partição, certamente a partição final pode conter regiões adjacentes com propriedades idênticas. Isto pode ser remediado realizando fusão. Para satisfazer as condições de segmentação estabelecidas, fundem-se apenas regiões adjacentes cujos *pixels* combinados satisfazem P , isto é, funde-se duas regiões adjacentes R_i e R_k apenas se $P(R_i \cup R_k) = \text{VERDADEIRO}$. Em resumo, pode-se executar o seguinte procedimento:

1. Dividir em quatro quadrantes desunidos qualquer região R_i para qual $P(R_i) = \text{FALSO}$.
2. Fundir quaisquer regiões adjacentes R_j e R_k para as quais $P(R_j \cup R_k) = \text{VERDADEIRO}$.
3. Parar quando não é mais possível fundir ou partir.

2.7.5 Operações de Morfologia Binária

Operações de morfologia se referem a uma família de operações que são executadas em uma imagem. Elas incluem muitas operações diferentes, podendo ser para imagens binárias ou em níveis de cinza, como engrossamento, dilatação, erosão, esqueletonização, abertura, fechamento e preenchimento. Estas operações são executadas em uma imagem para ajudar em sua análise assim como para reduzir as informações desnecessárias que possam estar presentes na imagem.

Espessamento: esta operação elimina pequenos buracos e falhas na borda de um objeto e pode ser usado para alisar sua superfície. Em um caso mais específico, por exemplo, pode ser usado para eliminar a imagem de borda “dentada” de um parafuso, gerada pela sua rosca. Isto é importante quando se deseja aplicar outras operações, como esqueletonização, num objeto: o espessamento inicial impede o surgimento de linhas perpendiculares ao eixo longitudinal do parafuso, que seriam geradas pela rosca assim que fosse aplicada a esqueletonização.

Dilatação: neste método, os *pixels* de fundo que estão conectados-de-8 ao objeto são mudados para *pixels* de objeto. Como resultado, uma camada é adicionada ao objeto cada vez que o processo é implementado. Devido à dilatação ser executada em *pixels* que são conectados-de-8 ao objeto, repetidas dilatações podem mudar a forma do objeto. Além disso, objetos

próximos em uma imagem podem encostar após alguns passes de dilatação assim como orifícios em um objeto podem ser preenchidos. Pode-se chegar ao ponto de objetos se fundirem em um só, impedindo a identificação dos objetos.

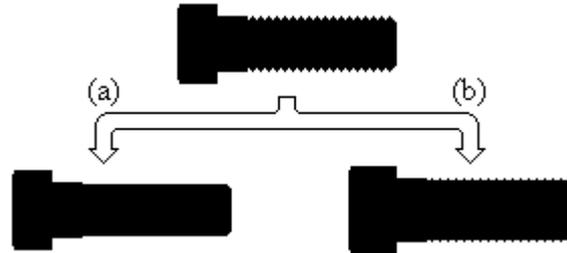


Figura 2.13 – Efeito aproximado de espessamento (a) e dilatação (b).

Erosão: Neste processo, *pixels* da imagem que são conectados-de-8 a um *pixel* do fundo são eliminados. Isto efetivamente elimina uma camada da borda do objeto cada vez que a operação é executada. Sabendo que a erosão remove *pixels* de toda a borda do objeto, este se torna cada vez mais fino a cada passe do processo. Entretanto, a erosão desconsidera todos os outros requerimentos da representação de forma. Irá remover uma camada de 1 *pixel* de espessura do perímetro, e de orifícios, do objeto até mesmo se a forma do objeto for eventualmente perdida. O resultado de muitos passes de erosão será a perda do objeto. Como resultado, este processo pode danificar irreparavelmente a imagem. Entretanto, ele pode ser usado para apagar objetos indesejados da imagem.

Esqueletonização: um esqueleto é uma linha que representa um objeto em que toda a espessura foi reduzida a 1 *pixel* em cada ponto. A esqueletonização é uma variação da erosão. Enquanto na erosão a espessura de um objeto pode ir a zero e este pode ser perdido, na esqueletonização, assim que a espessura do objeto atinge a de 1 *pixel*, a operação em todos os pontos pára. Do mesmo modo, enquanto na erosão o número de repetições poder ser escolhido pelo usuário, na esqueletonização o processo continua até o objeto atingir 1 *pixel* de espessura. Seu resultado final é uma linha (esqueleto) de um objeto, que é uma boa representação deste, muitas vezes melhor que sua borda. Outro ponto a ser notado, é que se um esqueleto for dilatado, muitas vezes o objeto resultante terá uma forma diferente do objeto original, assim como imagens de objetos que anteriormente eram mais finos que outros, terão a mesma espessura após um processo de esqueletonização seguida de dilatação.

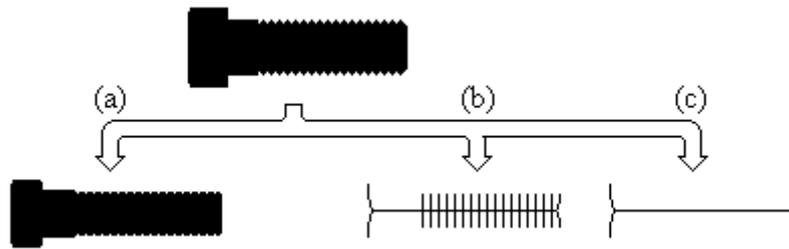


Figura 2.14 – Erosão (a) e esqueletonização sem (b) e com (c) espessamento prévio.

Abertura e Fechamento: processos que combinam erosão seguida de dilatação e dilatação seguida de erosão, respectivamente. Causam um limitado melhoramento de superfícies convexas de objetos. Podem ser usados como operação intermediária antes da esqueletonização.

Preenchimento: similar ao processo de dilatação, mas que se limita aos orifícios de um objeto. Estes são preenchidos por *pixels* do fundo.

2.8 Análise de Imagens

A análise de imagens é um conjunto de técnicas e operações que são usadas para extrair informação de imagens. Entre estas operações estão reconhecimento de objetos, extração de características superficiais, determinação de posição, orientação, tamanho e outras propriedades de objetos e extração de informações de profundidade da imagem.

Assume-se que técnicas de processamento da imagem já foram aplicadas à imagem ou que elas estão disponíveis para aplicação, se necessário, para melhorar e preparar a imagem para análise. As rotinas de análise de imagem podem ser usadas tanto para imagens binárias quanto em escala de cinza, bem como para mais de um propósito.

2.8.1 Reconhecimento de Objetos

Objetos podem ser reconhecidos em uma imagem através de características tais como: histogramas de níveis de cinza; características morfológicas como área, perímetro, número de orifícios, etc; excentricidade, comprimento, momentos de inércia, etc. Em muitos casos a informação extraída é comparada com informações conhecidas do objeto, que podem formar uma tabela de comparação. As características morfológicas básicas que são usadas em reconhecimento de objetos são a seguir descritas [Castleman, 1996]:

Média, valor máximo e mínimo de níveis de cinza

Usados para diferenciar objetos ou componentes em uma imagem. Separando a imagem em um número determinado de setores, objetos de interesse podem ser encontrados através da análise do histograma de níveis de cinza de cada setor e comparando a média ou os valores máximo ou mínimo com o valor previamente conhecido do objeto. Em alguns casos, apenas a presença de um determinado nível de cinza em um histograma pode ser suficiente para determinar a presença de um objeto.

Características morfológicas (perímetro, área, diâmetro e número de orifícios)

O perímetro pode ser facilmente encontrado aplicando-se uma rotina de detecção da borda e contando-se o número de *pixels* presentes. A área pode ser calculada através de técnicas de crescimento de região. O diâmetro de um objeto não-circular é definido como a máxima distância entre quaisquer dois pontos em qualquer linha que atravessa a área identificada do objeto.

Razão de Aspecto

É a razão entre a largura e o comprimento de um retângulo circunscrito em um objeto, como mostrado na Figura 2.15. Todas as razões de aspecto, exceto a mínima razão de aspecto são sensíveis à orientação. Portanto, a mínima razão de aspecto é geralmente usada para identificar objetos.

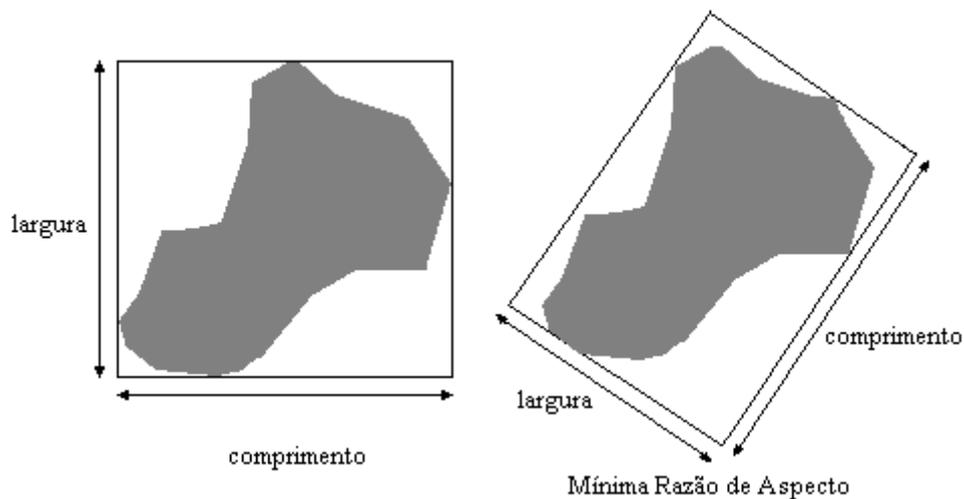


Figura 2.15 – Razão de Aspecto de um objeto.

Esbeltez

A esbeltez (E) é definida como uma das razões à seguir, onde p é o perímetro, d é o diâmetro e A é a área, do objeto:

$$\text{a.} \quad E = \frac{p^2}{A} \quad (16)$$

$$\text{b.} \quad E = \frac{d}{A} \quad (17)$$

2.8.2 Momentos

Os momentos de objetos em imagens utilizadas em visão robótica são de especial importância na análise de imagens pois são utilizados em várias tarefas de reconhecimento e interpretação, quer por cálculo simples ou por comparação em uma tabela pré-armazenada.

Considerando-se um objeto em uma imagem binária, onde *pixels* do objeto estão “ligados”, ou 1, e o fundo é representado por *pixels* “desligados”, ou 0. Tem-se a equação geral de momento:

$$M_{a,b} = \sum_{x,y} x^a y^b \quad (18)$$

Onde $M_{a,b}$ é o momento do objeto na imagem com índices a e b . As coordenadas de cada *pixel* “ligado” na imagem são representadas por x e y , com origem em um dos cantos da imagem. Primeiramente, uma rotina baseada na Equação (18) irá determinar se cada *pixel* pertence ao objeto (“ligado”) e, caso positivo, irá elevar as coordenadas da localização do *pixel* aos dados valores de a e b . O somatório desta operação sobre toda a imagem será o momento particular do objeto nos valores a e b . $M_{0,0}$ é o momento do objeto em $a = 0$ e $b = 0$. Isto significa que todos os valores de x e y estão elevados à potência de 0. $M_{0,2}$ significa que todos os valores de x estão elevados à potência de 0 e todos os valores de y estão elevados à potência de 2. É comum a aplicação de momentos que usem todas as combinações de índices entre 0 e 3 [Niku, 2001].

As distâncias x e y são medidas tanto dos eixos coordenados fictícios localizados na borda da imagem (x,y) ou são medidas dos eixos coordenados formados pelas primeiras linha e coluna da imagem. As distâncias são medidas contando-se o número de *pixels* do eixo até o *pixel* de interesse. O uso da primeira linha e coluna como eixos coordenados é mais comum, mas, como ambas as referências são usadas consistentemente, a escolha não é importante.

Baseado nestas definições, tem-se que todos os números elevados à potência de 0 são iguais à 1, então todos os x^0 e y^0 serão iguais à 1. Como resultado, o momento $M_{0,0}$ é o somatório de tantos 1's quantos *pixels* “ligados” há na imagem, fornecendo o número total de *pixels* “ligados”, o que representa a área do objeto. Em outras palavras, o momento $M_{0,0}$ é o mesmo que a área do objeto. Este momento pode ser usado para determinar a natureza de um objeto e para distinguí-lo de outros objetos que possuem áreas diferentes.

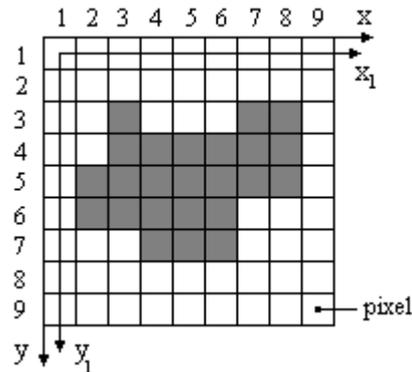


Figura 2.16 – Imagem exemplo para cálculo de momentos.

Similarmente, $M_{0,1}$ é $\sum x^0 y^1$, ou o somatório de $1 \times y$, que é o mesmo que o somatório da área de cada *pixel* multiplicado por sua distância do eixo x . Isto é similar ao primeiro momento da área em relação ao eixo x . A localização do centro da área de *pixels* 1 relativo ao eixo x pode ser calculado por [Gonzales e Wints, 1977]:

$$\bar{y} = \frac{\sum y}{\text{área}} = \frac{M_{0,1}}{M_{0,0}} \quad (19)$$

Também pela simples divisão dos dois momentos, pode-se calcular a coordenada \bar{y} do centro da área do objeto. Similarmente, a localização do centro de área relativo ao eixo y será:

$$\bar{x} = \frac{\sum x}{\text{área}} = \frac{M_{1,0}}{M_{0,0}} \quad (20)$$

Deste modo, um objeto pode ser localizado em uma imagem, independentemente de sua orientação. Logicamente, esta informação pode ser usada para localizar em um espaço de trabalho um objeto a ser manipulado por um robô.

Analisando outros momentos, $M_{0,2}$ é $\sum x^0 y^2$ e representa o segundo momento de área relativo ao eixo x . Similarmente, $M_{2,0}$ é o segundo momento de área relativamente ao eixo y . Deste modo, o momento de inércia de um objeto, como o da Figura 2.16, irá variar significativamente se o objeto rotacionar em torno de seu centro. Supondo-se que sejam calculados os momentos de área em relação ao eixo x , em diferentes orientações. Cada orientação cria um valor único, uma tabela de comparação (*lookup table*) que contenha esses valores pode ser usada para identificar a orientação do objeto. Logo, se uma tabela de comparação contendo os valores dos momentos de inércia de um objeto conhecido em diferentes orientações é calculada, a subsequente orientação do objeto pode ser estimada comparando seus segundos momentos com os valores na tabela. Quando um objeto translada dentro de uma imagem, estes momentos de inércia irão também mudar. Entretanto, se as coordenadas do centro de área do objeto são conhecidas, então, com uma simples aplicação do teorema de eixos paralelos, os segundos momentos do centro de área podem ser calculados independentemente de sua localização [Azuela, 2001]. Como resultado, com o uso de equações de momento, um objeto, sua localização e sua orientação podem ser identificadas. Em adição à identificação do componente, a informação pode ser usada para direcionar o robô para capturar a peça ou trabalhar sobre ela.

Outros momentos podem ser usados similarmente. Por exemplo, $M_{1,1}$ representa o produto de inércia da área e pode ser também usado para identificar um objeto. Momentos de ordens mais altas como $M_{0,3}$, $M_{3,0}$, $M_{1,2}$, etc., também podem ser usados para identificar objetos e suas orientações. Sendo dois objetos relativamente similares em forma, como na Figura 2.17(a), é possível que o segundo momento, áreas, perímetros ou outras características morfológicas dos objetos possam ser similares ou muito próximas entre si, logo não podendo ser úteis na identificação do objeto. Neste caso, uma pequena diferença entre os dois objetos pode ser exagerada através dos momentos de ordem mais alta, tornando a identificação do objeto possível. O mesmo vale para um objeto com uma pequena assimetria (Figura 2.17(b)). A orientação do objeto pode ser encontrada por momentos de alta ordem [Niku, 2001].

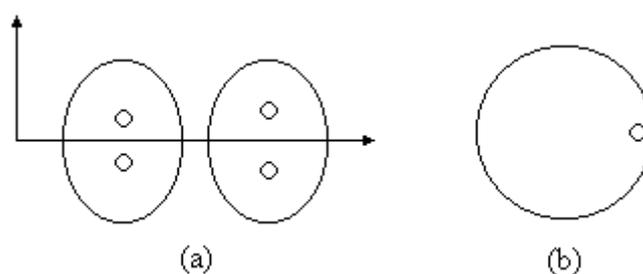


Figura 2.17 – Objetos com pequenas diferenças e assimetrias.

2.9 Sistemas de Controle e Transmissão de Sinais

Além dos procedimentos de captação, processamento e análise de imagem, têm-se os relacionados com a transmissão e recebimento dos resultados, variáveis e comandos gerados pelos dispositivos de visão industrial. Estes mecanismos de transmissão são a base do controle em tempo real na indústria em tarefas CAM, imprescindíveis para o funcionamento de dispositivos de visão robótica, que exigem respostas imediatas dos manipuladores a um ambiente de trabalho em constante processo produtivo.

O desenvolvimento dos controladores automáticos teve como objetivo facilitar o controle de processos em plantas complexas, que, no início do século XX, devido aos equipamentos mecânicos possuírem controles manuais e localizado próximo ao processo, demandavam grande esforço e coordenação. Foi através da aplicação de métodos e padrões analógicos de transmissão de dados, sendo um dos primeiros o padrão pneumático 3-15 *psi*, trabalhando em conjunto com processos instrumentados mecanicamente, que permitiu o surgimento do controle automático. Neste sistema, a pressão na linha de controle era proporcional ao valor da variável que se desejava transmitir. Um método mais eficiente para execução das mesmas tarefas foi desenvolvido mais tarde, através do envio de sinal elétrico de 4 a 20 mA, que se tornou um padrão utilizado até hoje [Murugesan, 2003], onde os dados eram transmitidos fazendo-se variar a corrente do circuito do sensor de forma proporcional à variação do valor da variável de controle.

Com o desenvolvimento da eletrônica e de dispositivos semicondutores, os sistemas analógicos puramente pneumáticos tornaram-se obsoletos. No início dos anos 80 os sinais elétricos analógicos provindos do chão-de-fábrica passaram a ser conectados a conversores analógico/digitais, para poderem ser processados por microprocessadores digitais. Os dados resultantes eram enviados para um conversor digital/analógico, sendo o sinal analógico utilizado para o controle de atuadores com posicionadores eletro-pneumáticos. Esta prática dominou o cenário fabril por mais de 10 anos, mantendo-se o conceito de sala de controle centralizada. No final da década de 80, surgiram no mercado os transmissores “inteligentes”, utilizando a transmissão de sinais digitais sobrepostos aos sinais analógicos, permitindo sistemas de diagnose remota [Mahalik, 2003].

O desenvolvimento de transmissores capazes de manipular toda a estratégia de controle usando blocos de funções integradas ao software e por comunicação ponto-a-ponto com equipamentos similares eliminou a necessidade do processamento centralizado das informações.

Através da rede de automação da planta, os controles foram levados para o chão-de-fábrica, mais próximos ao processo, trazendo de volta a estratégia de controle descentralizado.

2.9.1 Hierarquia de Sistemas de Automação Industrial

A automação em uma planta de manufatura pode ser implementada em diferentes níveis. Estes, podem ser classificados de diferentes formas, como o modelo de quatro níveis proposto por Boucher [Boucher, 1996], e que apresenta a seguinte hierarquia:

- Nível 1/Máquina: assegura que as operações da máquina correspondam à seqüência de operações planejada. Controle de CNC e robôs industriais;
- Nível 2/Célula de Trabalho ou Linha de Produção: decisões locais para coordenar as atividades de vários equipamentos, como definir seqüência de produção e manipulação de materiais;
- Nível 3/Chão-de-Fábrica: coordenar as atividades entre diversas células de manufatura ou linhas de produção. Tarefas como gerenciamento de materiais e controle de qualidade;
- Nível 4/Planta: automatizar o processo de tomada de decisões da empresa ou dar suporte de informação, como pedidos de clientes, compras e planejamento da produção.

A integração completa de todos estes níveis de processos de decisão, suportados por um sistema de informações computadorizado, é normalmente chamada Manufatura Integrada por Computador (CIM). Elementos importantes para essa integração são as redes locais, que permitem interligar diferentes equipamentos e controles. Com o objetivo de padronizar o desenvolvimento de redes de comunicação, o *International Standarts Organization (ISO)* estabeleceu um modelo de referência que descreve como as redes devem ser estruturadas em termos de camadas de responsabilidade. Este modelo é conhecido como o modelo *OSI (Open System Interconnect)*, e tem o propósito de dividir o problema da comunicação em uma série de subproblemas. O modelo *OSI* é estruturado em sete camadas, assim definidas [Mahalik, 2003]:

- Camada Física: assegurar o formato físico dos dados adequado ao seu transporte através de sinais elétricos pela rede;
- Camada de Enlace de Dados: estruturar dados com vista ao seu envio pela rede;
- Camada de Rede: estabelecer conexão e assegurar o correto encaminhamento da informação;

- Camada de Transporte: supervisionar o serviço da rede;
- Camada de Sessão: gerenciar a sessão de comunicação entre os nós envolvidos;
- Camada de Apresentação: providenciar para que os dois computadores envolvidos na comunicação se entendam;
- Camada de Aplicação: converter a informação do usuário ou de um formato digital adequado para sua transmissão ou converter os dados recebidos de uma transmissão em um formato inteligível para aplicação do usuário.

2.9.2 Redes Locais

O problema mais complicado na automação de chão-de-fábrica é o de interconectar máquinas e processos, principalmente devido à tendência histórica de cada fabricante de utilizar seu próprio protocolo de comunicação. Porém com o crescimento da demanda pela integração de máquinas em sistemas coordenados de produção, criou-se a necessidade de um meio comum sobre o qual os controladores dos equipamentos pudessem comunicar-se entre si e com computadores em níveis superiores.

Atualmente, os computadores em uma fábrica podem comunicar-se através de uma rede local (LAN), definida pela norma ISO 11898, sendo normalmente implementada em uma área limitada e de propriedade de uma certa organização [Buchanan, 2003]. Este sistema de comunicação privado permite a comunicação entre diversos dispositivos integrados na rede, a distâncias que variam de poucos metros até vários quilômetros. Os equipamentos conectados a rede incluem, entre outros, computadores, controladores lógicos programáveis, máquinas CNC, robôs industriais e dispositivos coletores de dados.

As topologias usuais na implementação das redes são os tipos anel, estrela e barramento, sendo esta última a mais amplamente utilizada. Na transmissão de sinais entre estações na configuração anel, a informação é transmitida a uma estação e quando o sinal não lhe é endereçado, esse é retransmitido para a estação seguinte. Um problema em alguma estação pode interromper toda a transmissão. No tipo estrela, centrada em um servidor, uma pane nesta estação central paralisa toda a rede. Já no tipo barramento, os sinais estão acessíveis a todas as estações, sendo apenas copiados pela estação à qual foram endereçados, sem passar por máquinas intermediárias e sem depender de um servidor central. Apenas uma quebra no barramento pode interromper a transmissão [Bedworth et al, 1991].

Para terem acesso aos sinais disponíveis na rede, as estações devem obedecer ao método de acesso empregado. O mais utilizado para o controle de transmissão, em conjunto com a

topologia de barramento, é Passagem de Permissão, ou *token*, sendo implementado na norma *IEEE 802.4 (Token Bus)* [Boucher, 1996]. Neste sistema apenas uma estação é autorizada a transmitir a cada momento, sendo esta permissão controlada através da passagem de um sinal de controle de estação para estação. Quando uma estação recebe o *token* ela pode transmitir mensagens por um determinado período de tempo, após o qual deve passá-lo para a próxima estação da seqüência. Utilizando este sistema se evita o problema da colisão na transmissão, e permite-se que o tempo máximo para o envio de uma mensagem seja previamente definido. Por esta razão o controle por *token*, ou variações deste sistema, tem sido amplamente utilizados em aplicações onde controle em tempo real é envolvido.

O envio de imagens de sistemas de captação para aplicativos de processamento e análise pode ser feito através de rede se o computador onde é executado o programa está em um ambiente diverso de onde se dá a operação de captação, com os resultados desta análise sendo repassados via rede ao controlador robótico se este está conectado a uma LAN. Em casos onde a captação e análise se dão no mesmo local, o envio das informações pode ser eficientemente realizado através do envio direto de sinais entre a captação/análise e análise/execução de tarefas.

3. DESENVOLVIMENTO DO SISTEMA

A união das características principais dos robôs industriais (tais como trabalho contínuo sem apresentação de fadiga, mantendo precisão e repetitividade, com conseqüente aumento de produção) com maior autonomia na execução de tarefas onde há variáveis continuamente em transformação (como identificação de objetos dentre um conjunto a ser analisado ou a identificação da posição de objetos que venham a se apresentar continuamente em posições aleatórias) tem sido assunto de grande interesse e pesquisa. Baseado no exposto no capítulo anterior, verifica-se uma parcela do grau de desenvolvimento e conhecimento acerca da geração e análise de imagens e transmissão de sinais necessários à execução de tarefas de visão robótica.

Sendo o foco deste trabalho o desenvolvimento e aplicação de um sistema de visão para tarefas de manipulação robótica industrial, pode-se dividi-lo em cinco módulos principais:

- Equipamento de captura de imagens;
- Iluminação;
- Programa de processamento de imagens;
- Interface PC/Controlador do robô;
- Programa de manipulação.

Esta divisão foi feita para tornar mais objetiva a execução do projeto. Apesar de independentes, os elementos operam de forma conjunta para formar o sistema de visão proposto.

Tendo em vista a simplicidade de especificação, projeto e implementação, procurou-se utilizar ferramentas facilmente encontradas no mercado ou disponíveis no Laboratório de Robótica. Observando-se, ainda, a aplicabilidade em um ambiente industrial, buscou-se utilizar equipamentos que se encontram em operação na indústria ou que poderiam vir a ser eficientemente empregados neste ambiente.

3.1 Equipamento de Captura de Imagens

3.1.1 Câmera

Baseado na prática comum, foram escolhidas as câmeras digitais, que são as mais utilizadas atualmente e são facilmente encontradas no mercado. Além disso, permitem uma rápida e eficiente interface com os demais sistemas que seriam empregados no projeto, gerando um sinal

digital facilmente captado por aplicativos de visualização. No início do trabalho, cogitou-se na utilização de uma *webcam*, pela sua simplicidade de operação e pelo fato de dispensar uma placa de captura de imagens, devido à sua conexão USB (*Universal Serial Bus*). Ressalta-se também o fato de esse tipo de dispositivo haver sido empregado em outros trabalhos acadêmicos sobre sistemas de visão em robótica com vistas à manipulação de objetos [Soares, 2004].

Durante a fase inicial do trabalho, optou-se pela utilização da *webcam* apenas para funções de desenvolvimento e teste do programa de processamento de imagens em sua fase de projeto. Para a implementação do sistema de visão e para garantir sua consistência como ferramenta de aplicação industrial, decidiu-se pela utilização de uma câmera CCD de circuito fechado, o que acarretou a necessidade de uma placa de captura de imagens. Após pesquisa no mercado, escolheu-se o modelo BW-2302EA da *Samsung*, uma câmera CCD monocromática. Possuindo entrada de tensão padrão EIA de 24V AC, utilizou-se uma fonte para CFTV da Nodaji, na tensão e corrente especificadas. Este modelo de câmera se encontra em uso em tarefas de pesquisa em sistemas de visão com análise de imagens e integrando sistemas de posicionamento de equipamentos de raios-x [LACTEC, 2004]. Utilizaram-se lentes SLA-124C *Auto-Iris* com montagem C, também da *Samsung*. A função *Auto-Iris* indica o fechamento automático da íris das lentes quando do desligamento da fonte de força, para proteção dos filtros ópticos e captadores de imagem. O conjunto câmera-lentes é ilustrado na Figura 3.1.



Figura 3.1 – Câmera CCD.

Tabela 3.1 – Dados técnicos do conjunto.

Câmera	
Dispositivo de Captura	537(H)x505(V) 1/3"CCD
Sistema de Varredura	2:1 (padrão EIA) 525linhas/60campos/ 30frames
Resolução	380(H) linhas de TV
Lentes	
Distância Focal	12 mm
Razão de Abertura	1:1,4
Formato da Imagem	6,4x4,8 mm

3.1.2 Placa de Captura

Para a aquisição do sinal de vídeo fornecido pela câmera, através de um cabo coaxial RGC 59, foi selecionada uma placa de captura compatível, dentre as disponíveis no mercado. A placa selecionada pela sua compatibilidade com a câmera e que cumpria os objetivos do trabalho foi a

VTV 2004. Placa muito usada em CFTV (Circuito Fechado de Televisão) baseado em PC, sendo compatível com *slot* PCI, captura com 30 fps e possui entradas para até 4 câmeras.

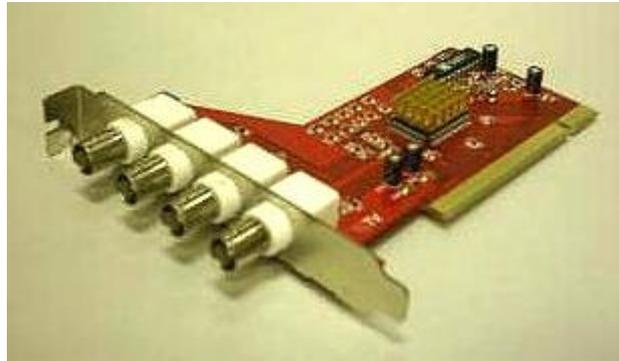


Figura 3.2 – Placa de captura de imagens.

Uma adaptação que se fez necessária durante a execução do trabalho foi a mudança do *driver* da placa. O original tornava a placa funcional com *softwares* de vigilância, mas não com programas multimídia gerados para o ambiente *Windows*. A simples mudança do *driver* tornou a placa reconhecível por demais *softwares* de captura/visualização e com programas gerados com linguagens de programação, como C e *Basic*, como o desenvolvido para este trabalho.

3.1.3 Suporte da Câmera

Analisados alguns esboços preliminares e observando o material disponível, definiu-se pela montagem do suporte do equipamento de visão com vigas de aço com seção em L. Caracterizam-se por serem leves e estáveis, além de já possuem furação, facilitando a montagem. Visando a facilidade de montagem e robustez do suporte, este foi fixado à mesa utilizada para a colocação dos objetos a serem manipulados, esta servindo também como uma base estável para o suporte. Sendo uma mesa industrial, já possui furação em sua estrutura de aço, podendo ser facilmente parafusada às vigas supracitadas.

Este suporte foi desenvolvido levando-se em conta as características peculiares do Laboratório de Robótica e apenas para fins didáticos, tendo sido especificado apenas para manter a câmera em uma posição acima da área de interesse e fora do alcance do manipulador e observando o material disponível para sua confecção. Ao se aplicar este sistema de visão em outro ambiente, como o industrial, devem-se levar em conta outros fatores, como vibrações e maquinário situado no entorno do robô além da disponibilidade de espaço, ao se projetar um suporte para a câmera.

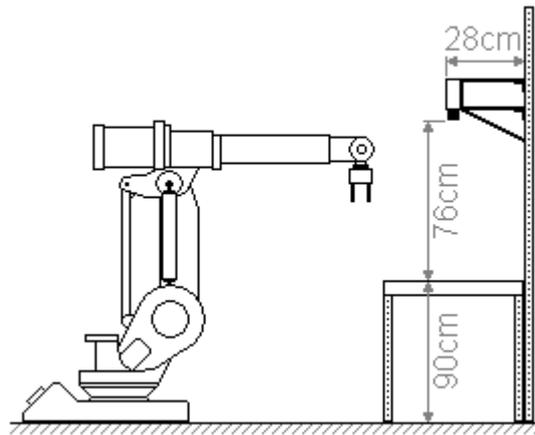


Figura 3.3 – Suporte da câmera.

3.2 Iluminação

Os tipos de sistemas de iluminação normalmente empregados em sistemas de visão robótica direcionados à manipulação são a iluminação direta e a traseira.

Apesar de a iluminação traseira permitir a geração de uma imagem ótima para este trabalho, devido à imagem ficar resumida a uma superfície iluminada e a sombra de um objeto (mais próxima de uma imagem preto-e-branco), desde o início considerou-se como a menos indicada, devido, principalmente, à dificuldade de implementação. Primeiramente, ela obriga que a superfície onde estarão os objetos a serem manipulados pelo robô seja translúcida ou transparente, para permitir a passagem da luz. Em segundo lugar, a iluminação do ambiente em torno da área de manipulação deve ser controlada, pois pode gerar interferência. Neste caso, a área de trabalho deve ser protegida por anteparos ou então a iluminação ambiente deve ser reduzida.

A escolha recaiu sobre um sistema de iluminação direta, que pode ser implementada sem alterações na iluminação do laboratório e sem a necessidade de anteparos ou superfície transparente. Como a superfície que é utilizada para manipulação no laboratório é clara, definiu-se a utilização de objetos de superfície escura a serem manipulados, o que, com um sistema de iluminação adequado, gera imagens muito próximas das obtidas com iluminação traseira.

Após a captação de algumas imagens sobre a área de trabalho, com o intuito de definir a melhor maneira de iluminá-la, constatou-se que as luminárias do ambiente fornecem iluminação suficiente e diretamente sobre a área de trabalho e praticamente sem a geração de sombras ou qualquer tipo de interferência.



Figura 3.4 – Iluminação da área de captura.

Em situações críticas para a iluminação, pode-se adaptar ao suporte da câmera, uma luminária do tipo fluorescente apontada diretamente para a área de interesse. No caso, como a qualidade da iluminação do laboratório demonstrou ser suficiente, eliminou-se a necessidade de utilização de mais um equipamento sendo simplificada a implementação do sistema.

3.3 Interface PC/Controlador do Robô

Obtidas as variáveis resultantes da análise das imagens, essas devem ser transferidas para o controlador do robô para que um programa próprio de manipulação as capture e execute a tarefa. Isto é feito na prática com o uso de interfaces de comunicação entre equipamentos e máquinas de processo em células de fabricação integrados por dispositivos dedicados ou redes de chão-de-fábrica.

No caso deste projeto, onde se necessita da comunicação entre um PC comum com o controlador ABB S4 do robô IRB1400, emprega-se o sistema de comunicação IRBCom, desenvolvido no Laboratório de Robótica. Caracteriza-se por ser uma interface *hardware* e *software* de fácil implementação, através da porta paralela do PC, e que permite ações de controle em tempo real [Bayer, 2004]. Dentro da hierarquia de redes, este dispositivo integra-se no Nível de Máquina.

A interface de comunicação do controlador do robô é uma placa E/S digital ABB DSQC 223, que apresenta 16 canais digitais de entrada e 16 de saída, destinada originalmente ao controle de dispositivos integrados à célula do robô, através de sinais discretos. Como interface de comunicação no PC, utiliza-se a porta paralela, que disponibiliza os dados em um formato que

pode ser transferido ao robô sem necessidade de manipulação. A flexibilidade do sistema caracteriza-se pelas rotinas desenvolvidas no *software* de interface no PC, compiladas em um formato DLL (*Dinamic Link Library*), empregados na comunicação e controle. As DLL's permitem a integração do sistema a qualquer programa desenvolvido para a plataforma Win32 [Bayer, 2004]. Já a implementação das rotinas no controlador é feita através de programação na linguagem nativa do mesmo, no caso, em ABB Rapid. Estas rotinas devem ser introduzidas no código fonte dos programas em que se deseja implementar a comunicação, como no caso do programa de manipulação desenvolvido.

A comunicação implementada é bidirecional, onde o PC e o robô tem igual capacidade de envio de dados. O funcionamento, porém, dá-se no modo Cliente/Servidor, tendo o PC como Cliente. Nesta configuração o fluxo padrão de dados é do PC para o robô, a transmissão no sentido inverso só ocorre sob solicitação do Cliente.

O *Programa Cliente* desenvolvido pelo usuário, que no caso deste trabalho é o Programa de Processamento de Imagens RobVis, obtém dados de alguma fonte, como o equipamento de captura de imagens, e, após processá-los e analisá-los, decide quais tarefas devem ser realizadas pelo robô e quais variáveis devem ser informadas. O programa chama rotinas disponíveis na *Camada de Aplicação*, informando a ação a ser realizada, bem como os dados e parâmetros necessários. Estas informações são repassadas para a *Camada de Enlace de Dados*, a qual verifica os sinais da *Camada Física*, aguardando o sinal de “pronto para a recepção” do robô. Os dados são convertidos para o formato de *bytes* e enviados seqüencialmente para a *Camada Física*, este envio inclui também todo o procedimento de sincronismo de transmissão. Cada *byte* é então transferido para a porta paralela, estando disponível para o circuito externo. Os sinais da porta paralela são recebidos pelo circuito da Placa E/S do robô, através de um dispositivo de *hardware* especialmente desenvolvido para esta aplicação, denominado Módulo IRBCom, o qual faz os ajustes de tensão necessários. No controlador do robô a *Camada de Enlace de Dados* do programa residente recebe a seqüência de *bytes* através da Placa E/S, recompondo os dados originais, os quais são repassados à *Camada de Aplicação* ou diretamente ao *Programa Servidor*, no caso o *Programa de Manipulação*, que analisa os dados recebidos, acionando os comandos necessários para executar a tarefa prevista.

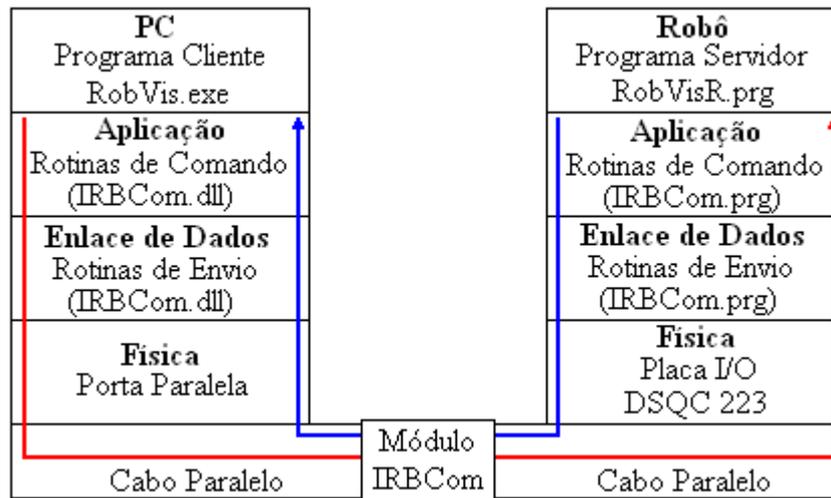


Figura 3.5 – Sistema segundo o modelo de camadas OSI.

3.3.1 Módulo Hardware IRBCom

O dispositivo de comunicação IRBCom apresenta características construtivas especialmente concebidas para interligar um PC via porta paralela. Este dispositivo é composto por uma série de acopladores óticos de alta sensibilidade modelo 4N33, os quais permitem a transmissão dos dados sem nenhuma conexão elétrica entre a porta paralela do PC e a Placa E/S do controlador do robô. Este sistema permite total segurança de operação, garantindo isolamento galvânico entre as duas interfaces para tensões da ordem de até 1,5kV.

No módulo IRBCom, as conexões são montadas em placas de circuito impresso e acondicionados em um gabinete com conectores externos. O painel frontal do gabinete possui *leds* indicadores de sinal em cada linha de comunicação, como ilustrado na Figura 3.6.



Figura 3.6 – Módulo IRBCom.

As conexões do sistema são feitas através de cabos paralelos de 25 vias com conectores DB25. No Módulo IRBCom utilizam-se dois conectores DB25 macho, um para conexão com o robô e outro para a conexão ao PC, pois esta configuração impossibilita a conexão errônea do cabo proveniente do robô diretamente na porta paralela do PC, o que causaria danos à mesma.

3.3.2 Módulo Software IRBCom

O módulo de comunicação IRBCom foi desenvolvido tendo em vista incorporar rotinas de transformação de dados gerados por um programa cliente para envio por rede e sua correta recepção e interpretação pelo destinatário, no caso, um programa servidor. Isto se enquadra, dentro do modelo *OSI*, nas funções especificadas para as camadas de enlace de dados e de aplicação.

Enlace de Dados

A Camada de Enlace de Dados é responsável pela estruturação de dados com vista ao seu envio pela rede, cabendo também um certo nível de controle de transmissão. É responsável de igual modo pela transição do formato de *bits* provenientes da rede, para um formato que possa ser entendido por níveis superiores. Constitui-se de rotinas pré-definidas para a transmissão e recebimento de dados, tendo implementação equivalente tanto no módulo do PC quanto no módulo do controlador do robô.

No módulo do PC estas rotinas estão encapsuladas em um arquivo chamado *IRBCom.dll*, o qual é compilado no formato DLL para a plataforma Win32. Este formato de arquivo permite a sua utilização por qualquer programa desenvolvido para o sistema operacional *Microsoft Windows*. Para o robô as rotinas foram escritas na linguagem nativa do controlador (ABB Rapid) e devem ser incluídas no código fonte do programa que pretende usar o sistema de comunicação.

Devido à baixa velocidade de processamento do controlador do robô, que pode bloquear a comunicação durante a execução do programa, implementaram-se sinais de sincronismo para garantir a máxima taxa de transmissão e evitar conflitos. Com um procedimento de sincronização viabiliza-se a transmissão, permitindo que esta seja interrompida e posteriormente retomada, por qualquer um dos dispositivos sem perda de dados.

A *Camada de Enlace* de dados define três categorias de dados, a saber:

- *Byte*: número natural, transmitido no formato de 8 *bits*;
- *Inteiro*: número inteiro, transmitido no formato de 16 *bits*;
- *Real*: número real, com 16 *bits* para a parte inteira e 8 *bits* para a parte fracionária.

Camada de Aplicação

É o nível mais próximo do usuário, implementando rotinas de interface entre o sistema de comunicação e o programa cliente desenvolvido pelo usuário. Trabalhando-se com as rotinas

padrão da Camada de Enlace de Dados é possível realizar a comunicação de forma efetiva. Visando facilitar a utilização, principalmente para as tarefas mais comuns, o sistema de comunicação é dotado de uma Camada de Aplicação, onde constam rotinas pré-configuradas e que satisfazem grande parte das necessidades básicas de um sistema de controle.

O protocolo de transmissão de dados é orientado a *byte*, sendo formado basicamente de um *byte* de comando seguido de uma série de *bytes* de dados. O número de *bytes* de dados é variável e definido pelo número e tipo dos parâmetros necessários para cada comando. A transmissão dos dados é feita chamando-se as rotinas da camada de Enlace de Dados ou rotinas específicas desenvolvidas nos próprios programas.

3.4 Programa de Análise de Imagens

Componente central do sistema de visão desenvolvido, o programa de processamento de imagens foi escrito em *Visual Basic*, devido não apenas ao seu difundido uso como ferramenta de criação de aplicativos, mas também à simplicidade da linguagem e facilidade de criação de interface gráfica para o usuário, sendo que o *software* utilizado na programação foi o *Visual Studio 6.0*. A este foi dado o nome de RobVis, acrônimo formado pelo termo empregado na literatura especializada *Robotic Vision*. Chegou-se a uma configuração que permaneceu como a melhor encontrada para a execução das tarefas propostas, estruturando-se o programa, e sua interface visual, em:

- Módulo de Captura;
- Módulo de Processamento;
- Módulo de Calibração.

Cada um deles executa tarefas específicas e, até certo ponto, independentes entre si, mas complementares. Podem operar independentemente, como no caso de calibração, onde o módulo de processamento não é ativado ou em análise de imagens, onde o módulo de calibração permanece desativado. As operações são interdependentes e complementam-se para o completo funcionamento do sistema.

3.4.1 Módulo de captura

É o módulo-raiz do programa. Nele são feitas a captura das imagens obtidas pelo conjunto câmera/placa de captura e sua apresentação na tela do PC. Também, é a partir dele que são ativados os módulos de processamento e de calibração, através de sua barra de menu.

Resumidamente, sua função básica é a geração de imagens para a operação dos outros módulos, onde são implementadas operações sobre as imagens captadas. Não sendo estes ativados, o módulo de captura funciona apenas como uma ferramenta de visualização de imagens. O programa foi desenvolvido para operar em configuração de vídeo 1024 × 768 ou superior. Uma representação da interface deste módulo é ilustrada na Figura 3.7.

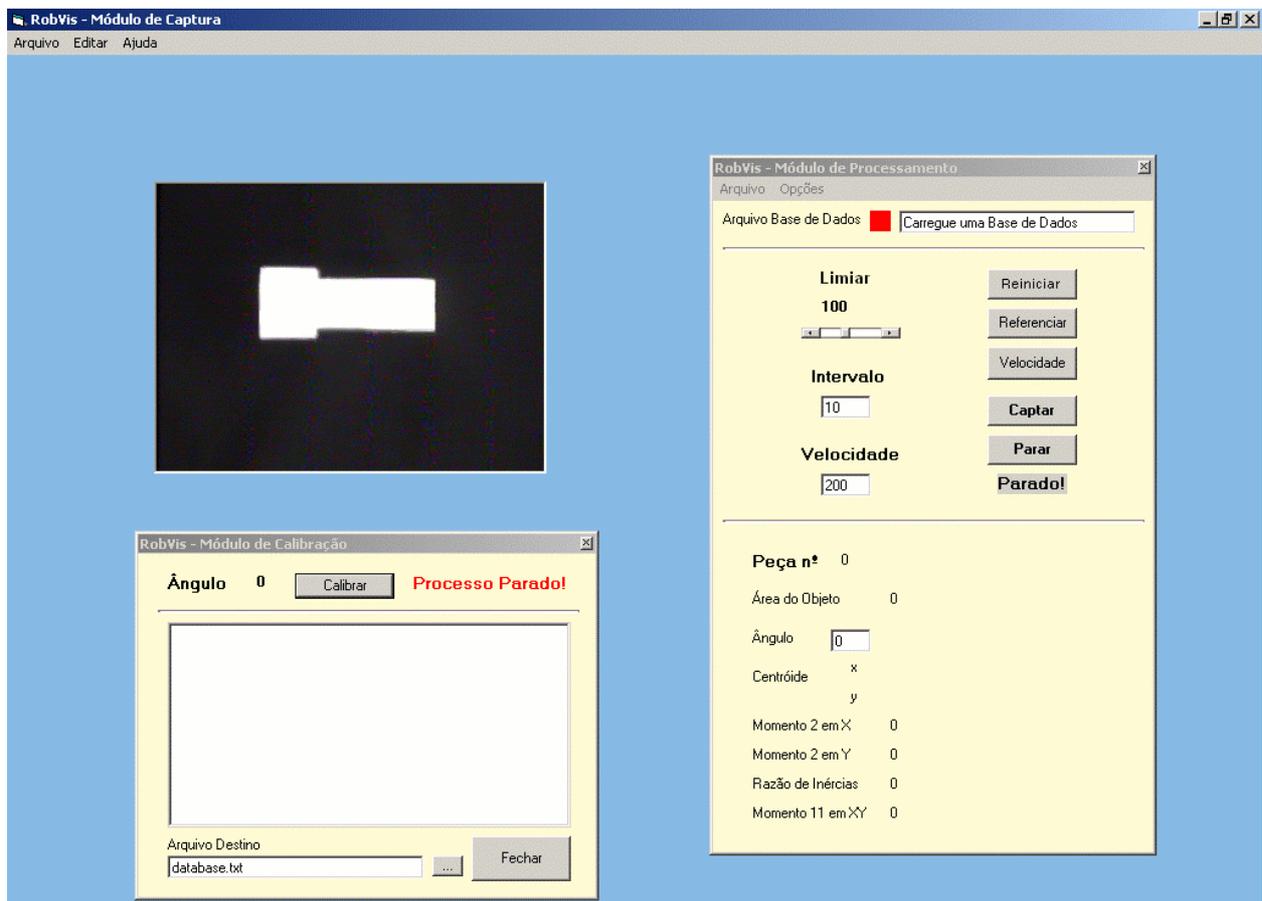


Figura 3.7 – Tela do módulo de captura com detalhes.

3.4.2 Módulo de Processamento

Módulo onde é feito todo o processamento e análise das imagens e geração de sinais para envio dos comandos e das variáveis obtidas com os procedimentos implementados para o robô. Também apresenta, na tela, os resultados obtidos ao usuário e permite que este especifique os parâmetros específicos para o caso em que o programa será empregado. Para uma melhor compreensão, a explanação sobre este módulo do programa está dividida em interface gráfica e procedimentos matemáticos, que serão apresentados no sub-capítulo 3.4.4 juntamente com os procedimentos do módulo de calibração. Sua interface gráfica se divide em:

Barra de Menu

Permite ao usuário selecionar se a análise de imagem será implementada em um sólido de revolução ou em um sólido cuja manipulação demande a informação de orientação. Neste caso, deve-se carregar uma base de dados que consiste em uma tabela de comparação (ou *look-up table*) em formato .txt previamente gerada pelo módulo de calibração para o componente específico a ser manipulado.

Também é possível selecionar operação em modo de simulação, que consiste apenas na execução das tarefas de processamento e análise da imagem em modo *off-line*, ou seja, sem envio de comandos para o robô. Este recurso é empregado em casos onde apenas se deseja verificar o funcionamento do programa, observando-se os resultados apresentados na tela, antes de se partir para a operação com o robô. Mantendo-se esta opção desabilitada, o programa gera comandos e envia variáveis ao controlador do robô.

Deve-se selecionar também se a operação se dará contra um fundo claro, e conseqüente objeto escuro, ou vice-versa. Desmarcando-se a opção “Fundo Branco”, a opção “Fundo Escuro” aparecerá na guia opções. A opção “Salvar Arquivo de Processamento” consiste na gravação de imagens binárias em formato .bmp para posterior avaliação. Serve para verificar se a captação está ocorrendo de forma satisfatória, assim como a limiarização das imagens sob o *threshold*, ou limiar, escolhido.



Figura 3.8 – Barra de menu do módulo de processamento.

Seção Base de Dados

Indica o nome do arquivo base de dados carregado para objetos com orientação ou se foi selecionado sólido de revolução, o que dispensa o uso de base de dados. A operação de análise não inicia sem a especificação da base de dados, sendo que se o usuário tenta iniciar a operação sem este procedimento é advertido para que faça a escolha. Do mesmo modo o indicador vermelho passa para verde quando o tipo de objeto é selecionado, para indicar visualmente ao operador que o programa já possui uma base de dados carregada.

Seção de Parâmetros e Comandos

Onde o usuário estabelece os parâmetros para operação das rotinas de processamento e análise da imagem. É composta pelos seguintes dispositivos, que estão detalhados na Figura 3.8:

- **Limiar:** uma barra de rolagem com valores de 0 a 256 para determinação do limite de limiarização. Deve ser ajustado conforme as condições de operação do sistema.
- **Intervalo:** intervalo de tempo entre captações para análise, em segundos.
- **Velocidade:** velocidade do robô em mm/s. Deve ser ajustada de acordo com as condições de operação.

- **Botão Reiniciar:** apaga os resultados apresentados na seção respectiva assim como retorna às configurações-padrão.

- **Botão Referenciar:** aciona o sinal 24 do módulo de comunicação, reconhecido pelo programa RobVisR.prg no controlador. O manipulador fica livre para ser movimentado pelo operador com o *joystick* de seu *teach pendant* para o ponto de referência área de captura / robô. Foi estabelecido como ponto de referência o canto superior-esquerdo da área de captura, sobre o qual deve estar posicionada uma peça das quais se vai manipular, com seu centróide aproximadamente sobre a referência. O manipulador deve ser posicionado como se fosse capturá-la, com o centro do efetuador sobre o centróide da peça.

- **Botão Velocidade:** aciona o sinal 23 do módulo IRBCom, seguido do valor de velocidade inserido na guia velocidade.

- **Botão Captar:** ativa o processo de captura de imagens seguido de processamento e análise. Este processo é executado respeitando o intervalo especificado.

- **Botão Parar:** interrompe o processo.

- **Guia de Status:** indica se não há processo em andamento ou se está sendo executado. Neste caso, um cronômetro indica o tempo decorrido entre procedimentos de captura.

Seção de Resultados

O usuário não tem controle sobre esta área, com exceção de poder zerar os resultados através do botão reiniciar. Esta seção se destina a mostrar os resultados da análise das imagens à medida que vão sendo executadas. É composta de:

- **Número da Peça:** indica quantas imagens já foram processadas dentro de uma seqüência de captação.

- **Área do Objeto:** indica a área do objeto presente na imagem captada em termos de *pixels*. É dependente do limiar adotado.

- **Ângulo:** orientação do objeto calculado através de seus momentos de inércia comparados aos gravados na base de dados carregada. Estes são peculiares de cada objeto e a forma de sua seção transversal. Sendo um sólido de revolução, o ângulo apresentado é sempre zero.
- **Centróide:** apresenta a posição do objeto sobre o plano de trabalho em relação ao centro da base do robô, em mm.
- **Momentos:** indica os momentos utilizados na determinação da orientação do objeto.

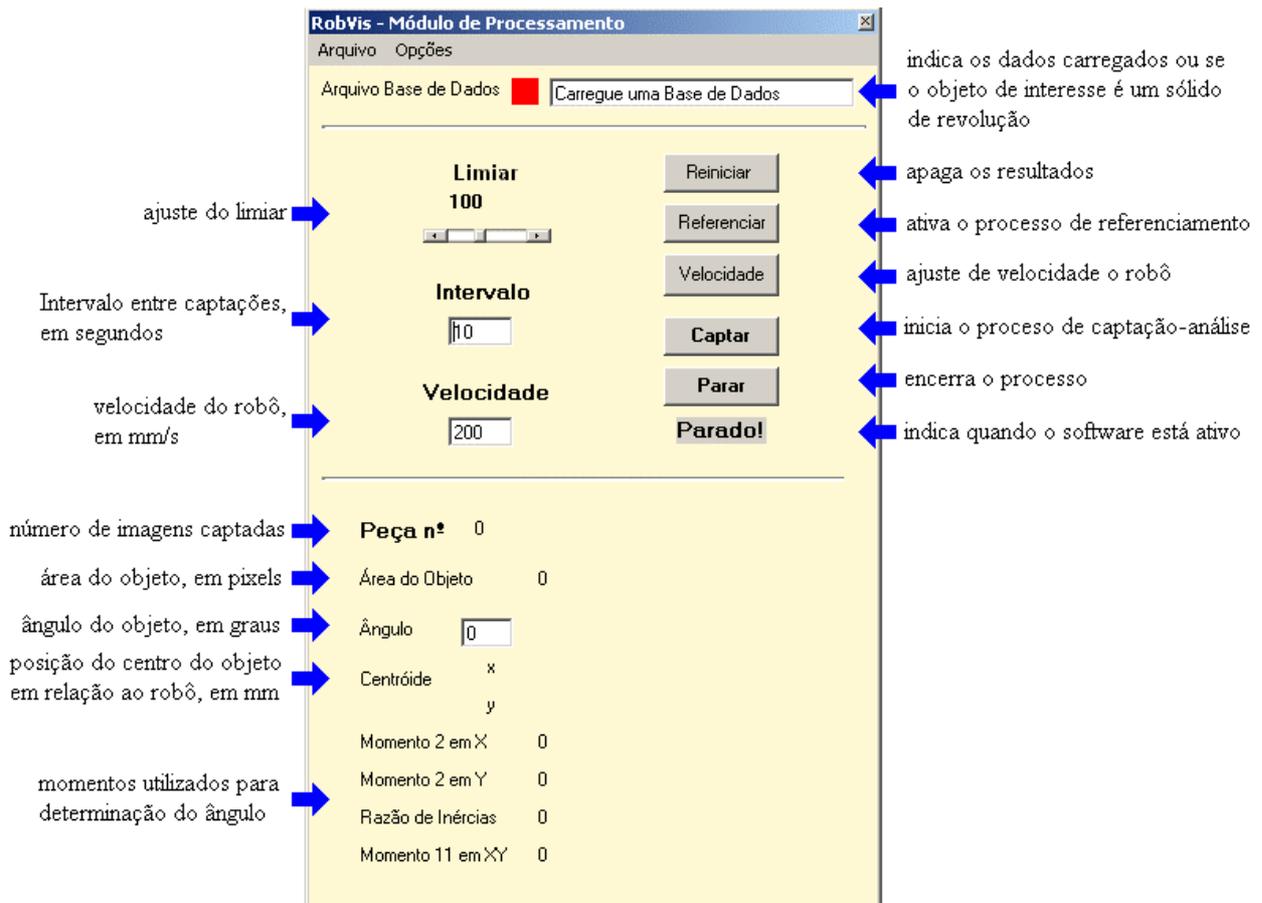


Figura 3.9 – Interface do módulo de processamento, com observações para suas funções.

3.4.3 Módulo de Calibração

É o módulo que deve ser ativado para gerar a tabela de comparação necessária para a determinação da orientação pelo módulo de processamento. Cada sólido, dependendo de sua geometria, apresenta alguns momentos de inércia dependentes da orientação assumida e são estes os parâmetros gerados pelo módulo de calibração para posterior comparação. Na sua execução, o operador ativa a seqüência de calibração e executa o procedimento apresentado na tela, que consiste em orientar, sob a área de captura de imagem, o objeto de interesse nos ângulos indicados na tela para que sejam atrelados a estas orientações valores salvos em um documento

.txt, posteriormente chamado pelo módulo de processamento. Estes valores serão discutidos na sub-seção 3.4.4. O referido documento, pode ser salvo com o nome e no diretório mais convenientes ao usuário, ficando armazenado para, em se necessitando posteriormente, voltar a empregar o mesmo sólido sem necessidade de nova calibração. Sua interface está representada na Figura 3.10.

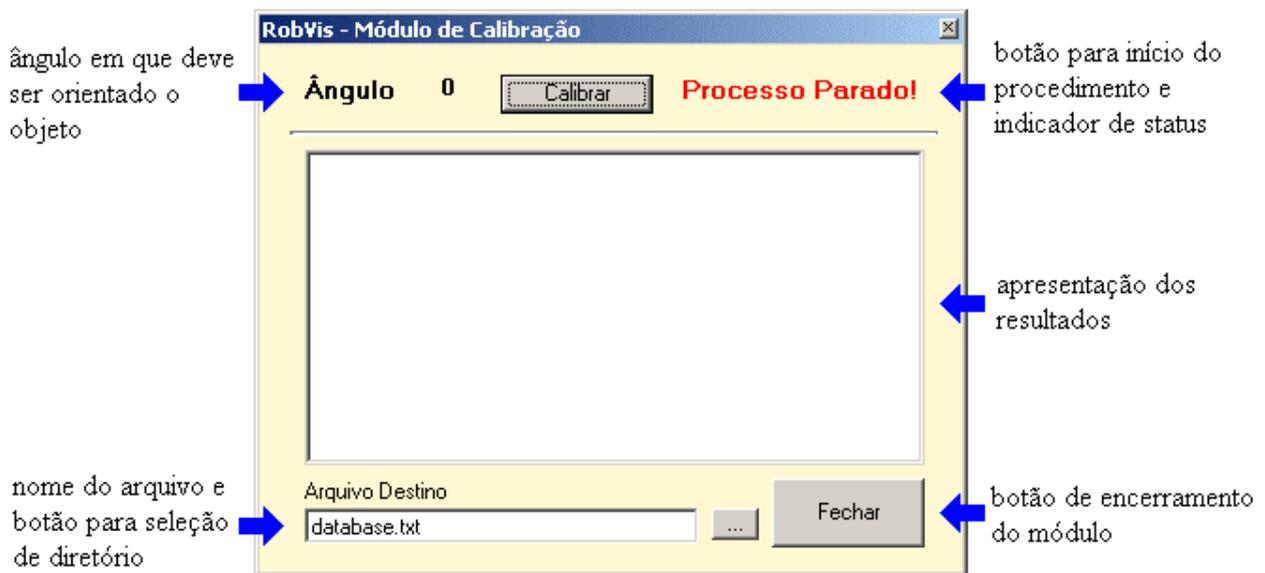


Figura 3.10 – Interface do módulo de calibração, com observações para suas funções.

3.4.4 Procedimentos Matemáticos

Nas rotinas implementadas nos módulos de processamento e calibração foram utilizados alguns métodos matemáticos introduzidos no Capítulo 2, selecionados dentre os expostos por mais eficientemente processar os resultados necessários à execução das tarefas propostas.

Limiarização

Como já descrito, a limiarização é uma das principais tarefas da visão industrial, por transformar imagens complexas em uma combinação de 1's e 0's ou preto e branco. Consiste em comparar o nível de cinza, que pode estar entre 0 e 255, de cada *pixel* da imagem independentemente com uma função lógica de maneira que resultem *pixels* com intensidades apenas 1 ou 0, dependendo de em que faixa da função estiver o valor original. A função empregada para fundos da imagem claros e objetos escuros é a seguinte:

$$g(x, y) = \begin{cases} 1 & \text{se } f(x, y) < T \\ 0 & \text{se } f(x, y) \geq T \end{cases} \quad (21)$$

Onde $g(x,y)$ é o valor de intensidade resultante do *pixel*, $f(x,y)$ é a intensidade original e T é o valor limite de intensidade. Como resultado, partindo-se de um limite T tal como 100, *pixels* escuros, próximos de 0, tendem a ser rotulados como sendo 1 (preto, “ligado” ou “objeto”) e quanto mais claros e acima do limite estabelecido, como sendo 0 (branco, “desligado” ou “fundo”), ficando isolados os *pixels* “ligados” e em condições de serem analisados. No entanto, se a tarefa será executado em um fundo escuro e com objeto claro, a função lógica fica:

$$g(x, y) = \begin{cases} 1 & \text{se } f(x, y) > T \\ 0 & \text{se } f(x, y) \leq T \end{cases} \quad (22)$$

Neste caso, ocorrerá o inverso do anterior, onde *pixels* escuros do fundo serão “desligados” e os demais “ligados”, ficando aptos para análise.

Determinação da Posição

Limiarizada a imagem e separados os *pixels* passíveis de serem analisados, implementa-se a contagem de *pixels* da área total da imagem, mais o número de *pixels* nos eixos virtuais x e y da imagem, e da área do objeto para determinação de sua posição. Isto é feito através das seguintes equações:

$$\bar{y} = \frac{\sum y}{\text{área}} = \frac{M_{0,1}}{M_{0,0}} \quad (23)$$

$$\bar{x} = \frac{\sum x}{\text{área}} = \frac{M_{1,0}}{M_{0,0}} \quad (24)$$

Onde o momento $M_{0,0}$ é o somatório de tantos 1's quantos *pixels* “ligados” há na imagem, fornecendo o número total de *pixels* pretos, o que representa a área do objeto. Em outras palavras, o momento $M_{0,0}$ é o mesmo que a área do objeto. Já $M_{0,1}$ e $M_{1,0}$ representam os somatórios dos *pixels* pretos multiplicados por suas distâncias em relação aos eixos x e y de imagem, respectivamente.

Este procedimento independe da geometria do objeto sobre o qual estiver sendo feita análise da imagem e fornece a posição do centróide dos *pixels* dentro da imagem o que corresponde a dizer que fornece o centróide do objeto dentro da área de captura de imagem, bastando referenciá-la em relação ao manipulador para se obter a posição do objeto no espaço em relação às coordenadas do robô. O procedimento de referência já foi apresentado e será tratado com mais detalhe no Capítulo 4, na descrição operacional do sistema.

Determinação da Orientação

Para sólidos de revolução, basta a informação de posição para que o manipulador esteja apto a capturá-los; neste caso, o operador do programa seleciona a opção correspondente a sólido de revolução e o programa envia sinal indicativo correspondente, mantendo a ferramenta sempre com ângulo de rotação 0° . Em geral, objetos tratados em tarefas de manipulação industrial apresentam geometrias que exigem o conhecimento da orientação de seu eixo de simetria ou de algum eixo que atravesse simetricamente a peça e que possa ser destacado para este procedimento.

O meio escolhido para a determinação de orientações neste trabalho foi o da criação da tabela de comparação através de momentos sensíveis à mudança de orientação do objeto. Neste caso os momentos $M_{0,1}$ e $M_{1,0}$ são invariantes quanto à orientação, o que não se aplica a momentos de mais alta ordem como $M_{0,2}$ e $M_{2,0}$, que consistem no somatório dos *pixels* multiplicados pelas suas distâncias ao quadrado dos eixos x e y , respectivamente. Neste caso, os momentos são dependentes da orientação, mas também da distância em relação aos eixos da imagem, o que se resolve calculando-os em relação a eixos auxiliares que têm origem no centróide do objeto. Calculados estes valores, para orientações conhecidas, cria-se uma tabela de comparação atribuindo-se a ângulos conhecidos valores de momento que um objeto de determinada geometria deve apresentar. Para simplificar o processo, ao invés de se fazer a comparação através de dois valores de momento, utiliza-se sua razão R , na forma que segue:

$$R = \frac{M_{0,2}}{M_{2,0}} \quad (25)$$

Tanto estes momentos como sua razão apresentam simetria de resultados de 0° a 90° em relação aos obtidos de 90° a 180° . Para diferenciar estes quadrantes, buscou-se utilizar um fator de diferenciação que poderia ser um sinal negativo. Os momentos $M_{0,3}$ e $M_{3,0}$ fornecem este sinal mas são muito sensíveis à ruídos na imagem por terem as distâncias elevadas ao cubo, acarretando erros excessivos. Como solução foi empregado o momento $M_{1,1}$ que consiste no somatório da distância de determinado *pixel* do objeto em relação ao eixo x que passa pelo seu centro multiplicado pela sua distância em relação ao eixo y que passa pelo seu centro. Ele fornece o sinal que diferencia se o objeto está rotacionado entre 0° e 90° (similarmente entre 180° e 270°) ou entre 90° e 180° (de forma similar entre 270° e 360°).

Após obtidos estes resultados pelo módulo de calibração eles são salvos em um arquivo .txt que deve ser nomeado pelo usuário para ser ativado pelo módulo de processamento quando se

empregam objetos de geometria igual à do que gerou a tabela. Os resultados são então inseridos em funções lógicas, da forma como segue:

$$se\ a < R \leq b\ e\ sinal = c,\ ang = d \quad (26)$$

Onde R é a razão de momentos, $sinal$ é o sinal positivo ou negativo do valor resultado de $M_{1,1}$ (ambos calculados durante a operação de análise de imagem), a e b são valores de razões de momentos conhecidos para determinados ângulos, c é o sinal positivo ou negativo do valor resultado de $M_{1,1}$ (ambos tomados durante a calibração) e d é o ângulo conhecido assumindo que a faixa $a - b$ e o sinal c correspondem e foram calculados para este ângulo.

Como exemplo: durante a calibração no ângulo de 45° determinou-se que este apresentava razão de momentos 3 e sinal negativo. Se 30° apresentava razão 2 e 60° , 4, durante a operação de análise assume-se que ângulos próximos de 45° cairão num faixa que vai de 2,5 a 3,5 ($2,5 < R \leq 3,5$) e terão sinal negativo ($sinal$ “-”). Se, em algum momento durante a operação do sistema, o programa encontrar $R = 3,2$ e $sinal$ “-”, por exemplo, o robô será instruído a orientar a ferramenta a 45° para a captura do objeto na posição (x,y) informada.

Observa-se que o programa de calibração está preparado para fazer a tomada de valores de 15° em 15° entre 0° e 165° , o que é suficiente, devido à simetria de resultados nos demais quadrantes. Essa calibração permite que o robô possa apanhar objetos orientados em qualquer ângulo, aproximando sua orientação para os ângulos intercalados de 15° , cabendo ao operador, posicionar o objeto de interesse nos ângulos solicitados e na ordem que é indicada na tela do módulo de calibração. Para cada ângulo, o módulo de calibração faz 10 capturas de imagem, em intervalos de 2 segundos, e conseqüentes análises. São feitas várias capturas para cada ângulo visando amenizar uma eventual tomada de valores errônea devido a algum ruído significativo presente em alguma captura de calibração. Após análise para cada uma das 10 capturas, o programa faz a média dos resultados, que é armazenada na tabela de comparação como sendo o valor de referência para aquele ângulo.

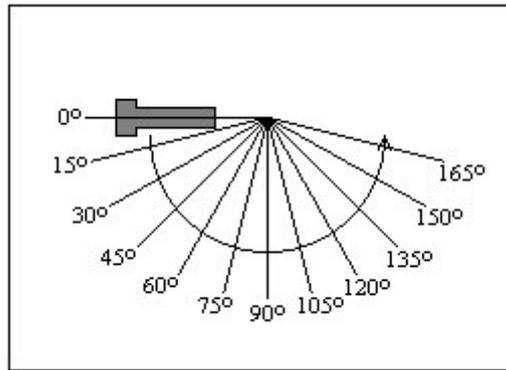


Figura 3.11 – Área de captura com os ângulos para calibração.

3.5 Programa de Manipulação

É o programa de produção que reconhece e capta as informações oriundas do programa de processamento de imagens e as integra às suas rotina de manipulação de componentes. Sendo o robô empregado no desenvolvimento deste trabalho um modelo ABB IRB1400, o programa de manipulação foi desenvolvido em linguagem ABB Rapid. Para reconhecimento de dados, foram adicionadas as rotinas específicas em seu código fonte.

Diferente de um programa comum de manipulação, dados como a posição do objeto a ser capturado não estão incluídos no código fonte do programa. Ao invés disso, apresenta lacunas a serem preenchidos pelas coordenadas enviadas pelo PC. O programa, denominado *RobVisR.prg* e executado no controlador do robô, tem como elemento central um laço que faz a leitura do comando, o identifica e chama a rotina solicitada, ficando a mesma responsável para receber seus parâmetros e executar a tarefa para a qual foi programada. Concluída a tarefa, o programa retorna ao laço central e chama *ReceiveByte* para receber o próximo comando e, caso o PC não tenha nenhum comando a ser enviado, o controlador permanece aguardando. Do mesmo modo, no intervalo de tempo em que a tarefa está sendo executada, o controlador fica bloqueado para outros comandos. Nesta situação, o PC verifica que o controlador está indisponível e mantém-se aguardando até a liberação, quando então envia novo comando. A relação entre os componentes do sistema pode ser vista na Figura 3.12.

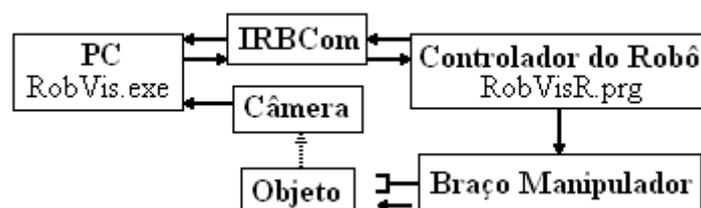


Figura 3.12 – Comunicação entre os componentes do sistema.

A estrutura do programa consta de uma raiz PROC Main, padrão da linguagem, que executa os *loopings* no aguardo de comandos provenientes do PC e chama as rotinas (PROCs) secundárias, específicas para execução das tarefas de manipulação, assim que um comando disponível no módulo de comunicação é acionado pelo programa RobVis. Suas rotinas secundárias são:

- *Place*: acionado sempre quando o programa é carregado pelo controlador do robô. “Questiona” o usuário se este deseja manter o ponto de descarga padrão ou se há um novo ponto de descarga para ser utilizado. O programa libera o braço do manipulador para ser movido pelo operador, que deve posicioná-lo no novo ponto de modo que o efetuado fique em uma posição como se fosse capturar um objeto.

- *DoHome*: rotina acionada pelo Programa Cliente do PC para que o usuário posicione o manipulador no ponto de referência. Do mesmo modo que a rotina Place, o manipulador fica livre e deve ser posicionado como se fosse capturar um objeto no canto superior esquerdo do campo de visão apresentado na tela do PC para calibração da posição espacial relativa do campo de visão em relação ao manipulador.

- *SetSpeed*: rotina acionada quando solicitado pelo usuário a partir do PC. Altera a velocidade do manipulador de 200 mm/s para qualquer velocidade desejada pelo usuário, se houverem novas condições de tempo a serem cumpridas. Existe a limitação de que o usuário deve definir velocidades com valores de 100 em 100 a partir de 500 mm/s, visto que o controlador só aceita velocidades em intervalos de 50 mm/s até V500.

- *Move*: é a rotina principal de manipulação. A partir do momento em que o programa do PC inicia a captura, processamento e análise das imagens e informa os resultados para o controlador, a rotina Move passa a ser acionada em seqüência onde são inseridos os valores de posição x, y e orientação plana do objeto, sendo acionada a seqüência de movimentos de aproximação, captura, transporte e descarga do objeto de trabalho.

Para acionamento dessas rotinas, é utilizada uma rotina do tipo *ReadCommand* que, através de um *ReceiveByte*, recebe o sinal do controlador e o define como sendo para referência, se o sinal recebido tiver valor 24, para ajuste de velocidade, se o valor for 23, ou de movimentação, se for 10. Estes valores são os padrões estabelecidos pelo programa IRBCom. Definido o tipo de comando que está em andamento, o programa passa para a leitura dos valores enviados em seqüência pelo PC utilizando o procedimento *ReadReal*, uma vez para o comando *SetSpeed* e três vezes pelo comando *Move*. O comando 24 *DoHome* simplesmente faz com que o controlador libere o manipulador para movimentação pelo usuário através do *joystick* presente no *Teach Pendant* do controlador S4.

Um exemplo de operação é a do comando *Move*, que tem seu ciclo através da seguinte seqüência:

- controlador informa estar livre para receber comandos, chamando *ReceiveByte*;
- Programa Cliente no PC chama a rotina *Move* da Camada de Aplicação, indicando também os parâmetros necessários;
- a rotina *Move* utiliza *SendByte* para enviar o valor 10, que é o código do comando;
- o Programa Servidor do controlador do robô recebe o *byte* e reconhece o comando, acionando a rotina *Move*, disponível na sua Camada de Aplicação;
- rotina chamada solicita a leitura dos parâmetros necessários, chamando *ReceiveReal* três vezes para receber os valores de x, y e o ângulo de orientação;
- rotina realiza a tarefa utilizando os parâmetros recebidos associados com comandos de movimento para mover o efetuador para a posição x, y na orientação indicada e levar a peça capturada até o ponto de descarga pré-definido, passando por pontos de trânsito, conforme a Figura 3.13;
- terminada a tarefa retorna ao laço principal, onde o controlador solicita um novo comando chamando *ReceiveByte*.

Note-se que durante a operação normal do sistema de visão, ou seja, em modo corrido de captura e análise de imagens com conseqüente geração de comandos de captura e deposição de objetos, o laço sempre recebe um novo comando *Move*, até o operador solicitar encerramento do ciclo através do Programa Cliente no PC, quando o *looping* fica livre para receber os demais comandos que possam ser gerados.

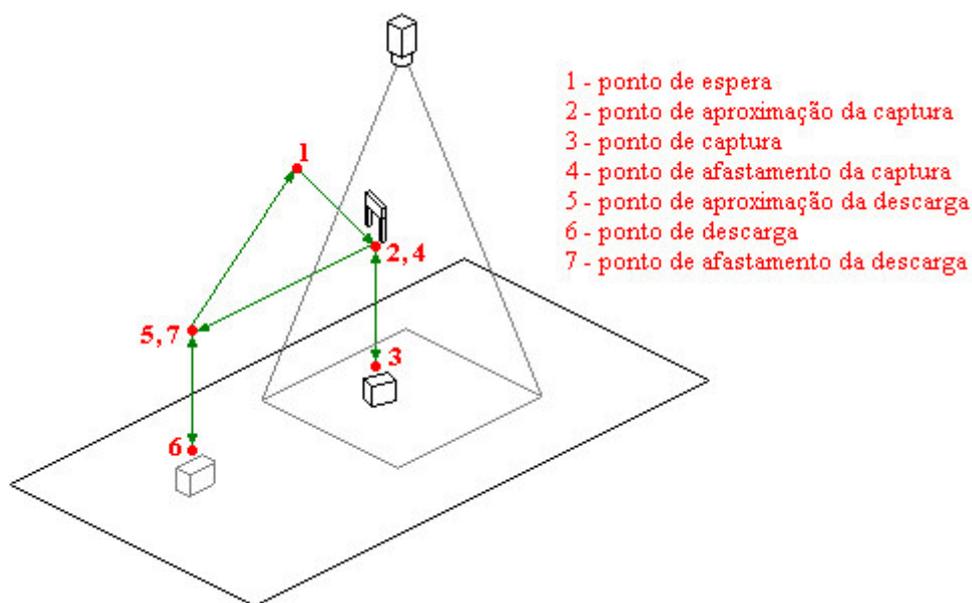


Figura 3.13 – Trajetória seguida pelo manipulador.

4. AVALIAÇÃO EXPERIMENTAL

Para validação do sistema de visão desenvolvido, executaram-se testes experimentais em laboratório, seja para avaliar a funcionalidade da integração dos componentes ou sua praticidade, eficiência e conseqüente aplicabilidade. Foram utilizados dois tipos de ferramentas adaptadas ao robô, uma garra Schunk e um gerador de vácuo Schrader Bellows, aqui chamado de ventosa, equipamentos direcionados à aplicação em robótica. Ambos têm acionamento pneumático e são comandados por sinais elétricos gerados pelo robô. No comando da garra utiliza-se um sinal para abertura e outro para fechamento e na ventosa, apenas um para gerar vácuo, sendo desligado para desativá-la. Esta peculiaridade foi observada na geração do programa RobVisR, que aciona rotinas separadas de ativação de sinal dependendo da ferramenta empregada. Nos testes com sólidos de revolução, foi empregada a ventosa e para os demais, a garra.

Como corpos de prova para a avaliação do programa, foram empregados 3 sólidos: um de formato cilíndrico, um bloco de seção retangular e uma peça com seção T. Isto visando verificar como se comporta a geração de tabelas de comparação para diferentes sólidos com orientação e sua posterior aplicação. Também foi feita avaliação da operação de sólido claro/fundo escuro e vice-versa, sendo os sólidos brancos com uma face pintada de tinta escura para contraste.

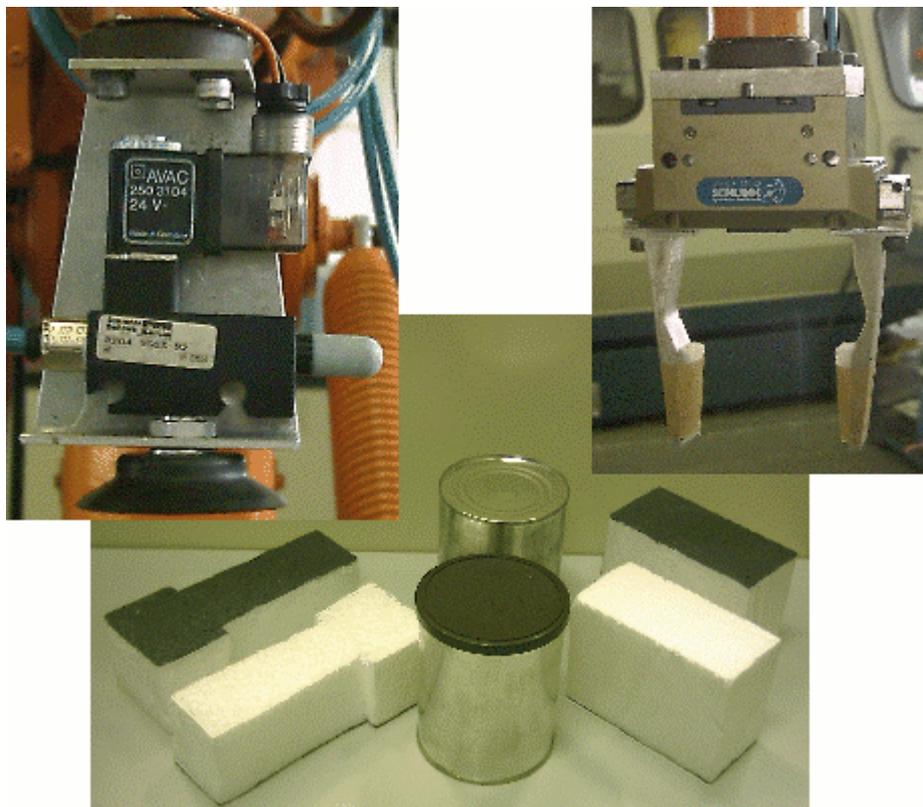


Figura 4.1 – Ferramentas e corpos de prova.

4.1 Desenvolvimento da Tabela de Comparação

A geração da tabela de momentos é o processo que antecede a operação do sistema quando se vai manipular sólidos orientados sobre o plano de trabalho. Deve-se manter sob a área de captura um gabarito ângulos de 0° a 165° , espaçados de 15° em 15° , com o eixo correspondente a 0° alinhado com o eixo horizontal da área de visão.

No aplicativo do sistema ativa-se o módulo de calibração a partir da barra de menu do programa RobVis. O primeiro passo é escolher o diretório onde deve-se salvar a tabela e seu respectivo nome. Como padrão, o aplicativo apresenta a pasta onde está o programa. Na sequência, pressiona-se o botão calibrar, ao que surge uma janela de diálogo indicando “posicione a 0° ”. Posicionando a peça alinhado com o eixo 0° e pressionando “OK”, o processo é iniciado, devendo ser executado seguindo os passos indicados na tela através de janelas de diálogo, que consiste em posicionar a peça de 15° em 15° . Para cada posição são feitas 10 tomadas em um intervalo de 2 segundos. Ao fim do processo, uma nova janela de diálogo indica o fim da calibração, podendo ser encerrado o módulo ou prosseguir com a calibração para outro sólido.



Figura 4.2 – Aparência do módulo e imagem durante calibração para 45° .

A vantagem do arquivo *.txt* gerado contra a criação de uma tabela temporária dentro do programa é a de poder ser armazenado para posterior uso, mesmo quando foram geradas outras tabelas. A Tabela 4.1 mostra exemplos de tabelas de comparação geradas para as geometrias utilizadas com as razões de momentos e o sinal diferenciador.

Tabela 4.1 – Tabelas de comparação para seção “T” e retangular.

Ângulo (°)	"T"		Retangular		Ângulo (°)	"T"		Retangular	
	Razão	Sinal	Razão	Sinal		Razão	Sinal	Razão	Sinal
0/180	140,5	-	269	-	180	140,5	+	269	+
15/195	295,5	-	406,5	-	195	295,5	+	406,5	+
30/210	678,5	-	746,5	-	210	678,5	+	746,5	+
45	1544	-	1425,5	-	225	1544	+	1425,5	+
60	3692	-	2592,5	-	240	3692	+	2592,5	+
75	7168,5	-	3797	-	255	7168,5	+	3797	+
90	7444,5	+	3850,5	+	270	7444,5	-	3850,5	-
105	4090	+	2657,5	+	285	4090	-	2657,5	-
120	1693,5	+	1468	+	300	1693,5	-	1468	-
135	710,5	+	795,5	+	315	710,5	-	795,5	-
150	307,5	+	427	+	330	307,5	-	427	-
165	146	+	281	+	345	146	-	281	-

4.2 Manipulação de Objetos com Orientação

Aberto o módulo de calibração, deve-se carregar um arquivo base de dados contendo os valores para a tabela de comparação. Isto é feito através de comando correspondente com uma ferramenta de busca de diretórios, devendo-se selecionar a tabela de comparação correspondente ao sólido a ser manipulado. Após a escolha, o módulo está livre para ter seus parâmetros estabelecidos para posterior emprego.



Figura 4.3 – Carregando base de dados.

4.2.1 Carregando o Programa de Manipulação

No comando do robô, a ativação do programa RobVisR pode ser feita diretamente de sua memória ou carregado a partir de disquete. Este, ao ser inicializado, questiona o usuário para manter o ponto de descarga pré-estabelecido ou escolher outro através da tela do *Teach Pendant* do controlador. Escolhendo a opção de mudar o ponto de descarga, o manipulador fica livre para ser movido por seu *joystick*. O usuário deve posicionar o sólido sobre o ponto que se deseja marcar como de depósito e então levar o manipulador, munido de efetuador, até ele, em condições de efetuar captura. Memorizada a posição através do menu do *Teach Pendant*, o ponto fica gravado no programa RobVisR. A Figura 4.4 ilustra o caso de memorização de um ponto de descarga com um bloco em T e utilizando uma garra como efetuador.

Há ainda a possibilidade de se efetuar esta mudança em modo *off-line*, através de rotina similar à descrita e gravando-se um ponto posteriormente inserido na raiz do programa em *ABB Rapid*, visto que este permanece passível de alterações até mesmo a partir de um simples editor de texto, como o bloco de notas do *windows*.

4.2.2 Referência do Campo de Visão

Este processo é importante por estabelecer uma relação de posição relativa do campo de visão com o robô e é o primeiro passo para o início da operação em ciclos de manipulação. A posição calculada para o objeto na imagem é transformada para coordenadas espaciais, sendo que a coordenada z tomada no momento da referência é assumida como sendo fixa para capturas e descargas de objetos.

O referenciamento é executado buscando-se posicionar o centróide do objeto o mais próximo possível ao vértice superior-esquerdo da linha que limita a área de captura de imagem. Ativando o botão “referenciar” no módulo de processamento, um sinal é enviado ao robô para que libere o braço, deixando-o livre para ser movido através do *joystick*, em procedimento similar ao descrito na subseção anterior. É um procedimento que requer atenção, pois um erro no referenciamento persiste durante o ciclo de operação, podendo acarretar tomadas fora de posição.

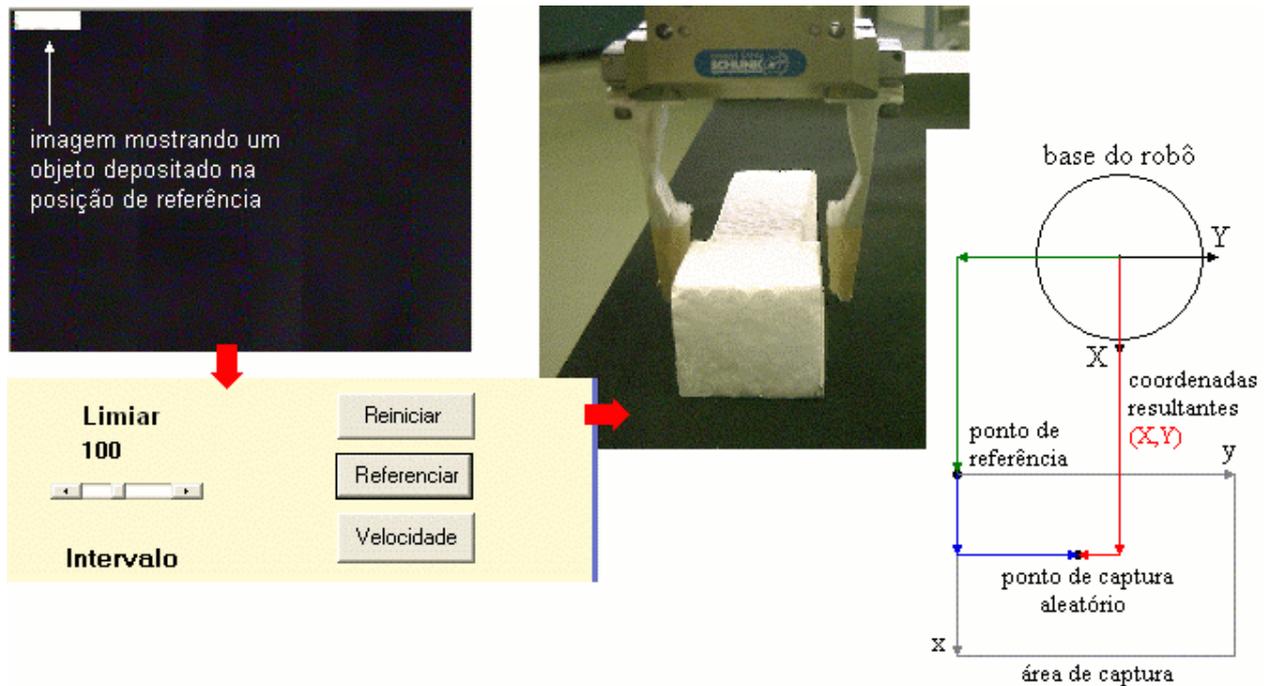


Figura 4.4 – Procedimento de referenciar o campo de visão.

4.3 Seleção de Parâmetros

A seleção dos parâmetros de operação reveste-se de importância na medida em que este procedimento personaliza a execução do sistema com as peculiaridades a situação em que se vai trabalhar. Deve-se ter atenção nos seguintes pontos:

- Modo de simulação: empregado apenas para verificação de funcionamento do programa em modo *off-line*, ou seja, sem envio de sinais. Por padrão do programa esta opção está selecionada, devendo-se desmarcá-la para operação normal;

- Fundo: para especificar condições de objeto claro/fundo escuro e vice-versa. A opção “fundo branco” está selecionada, devendo-se desmarcá-la, ao que é substituída pela opção “fundo preto”, quando se deseja operar nesta condição;

- Limiar: especifica o limiar de binarização da imagem, sendo dependente das condições ambientais e de iluminação, não havendo regra para estabelecimento de um valor adequado. Deve-se encontrar um limiar mais adequado para cada operação de manipulação. Pode-se, por exemplo, operar em modo de simulação e analisar os resultados apresentados na seção de parâmetros, além de gerar “arquivos de processamento”, disponível no menu “Opções”. Este salva imagens binárias, podendo-se verificar se o limiar assumido estabelece uma correta definição do objeto.

- Intervalo: depende das condições de operação, como a frequência em que é solicitada a manipulação de objetos e a velocidade em que o robô opera. É o intervalo de tempo, em segundos, entre análises de imagem sucessivas.

- Velocidade: depende de parâmetros similares aos descritos acima. É a velocidade do manipulador em mm/s. Após especificado um valor de velocidade, aciona-se o botão “Velocidade” que ativa o envio de um sinal seguido do número escolhido para o controlador. Recebido pelo programa RobVisR, este reconhece o sinal como indicativo de velocidade e lê o valor agregado, assumindo-o como a variável de velocidade constante nas linhas de comando. Caso o operador não defina a velocidade ao manipulador, o programa de manipulação assume 200 mm/s ao iniciar o ciclo. A Figura 4.6 ilustra a interface de definição de parâmetros.

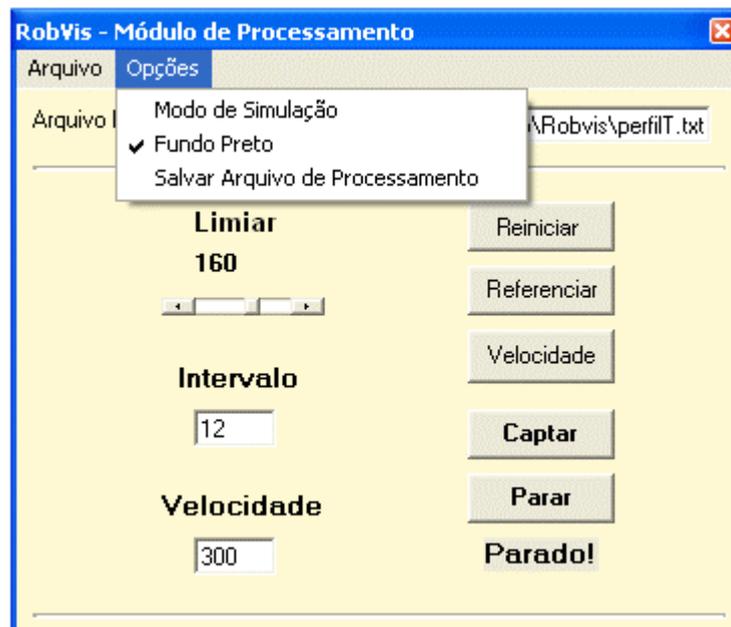


Figura 4.5 – Seleção dos parâmetros.

4.4 Ciclo de Operação

O ciclo operacional é ativado através do comando “Captar” do módulo de processamento. As imagens são captadas no intervalo especificado, sendo processadas e analisadas, e seus resultados enviados à unidade de comando. O programa RobVisR reconhece o sinal de movimento seguido de três variáveis: x , y (as coordenadas do objeto no plano de trabalho) e sua orientação. Estes são agregados às suas linhas de comando que movem o manipulador até a posição e orientam sua ferramenta para a captura. Na seqüência, é executada a trajetória pré-determinada, composta por pontos de aproximação e afastamento, até o ponto de descarga estabelecido, descarregando a peça e voltando até o ponto de espera fixo, onde o *looping* do

programa aguarda novos comandos. Os citados pontos de trânsito e espera estão localizados a distâncias seguras do suporte e da câmera além de posicionados de modo a garantir que o manipulador vai estar fora do campo de visão quando da captação. Este procedimento ocorre conforme o descrito no subcapítulo 3.5 e prossegue até que o operador selecione o comando “Parar”.



Figura 4.6 – Imagens captadas durante o ciclo de operação.

4.5 Operação com Sólido de Revolução

O sistema tem tratamento diferenciado para sólidos de revolução, bastando especificar “Sólido de Revolução” na base de dados do módulo de processamento, que assume orientação 0° para a ferramenta em todo o ciclo de operação. Define-se a referência do campo de visão pelo mesmo processo e a seleção de parâmetros conforme as condições já descritas, operando-se o ciclo do mesmo modo. A trajetória seguida pelo manipulador é a mesma ilustrada no subcapítulo 3.5.

A opção “Sólido de Revolução” também se presta à operação de sólidos orientados quando se garante que estes serão sempre capturados e depositados com o mesmo ângulo, não sendo

necessário orientar a ferramenta. Um exemplo de aplicação é ilustrado na Figura 4.8, onde o efetuador é uma ventosa.

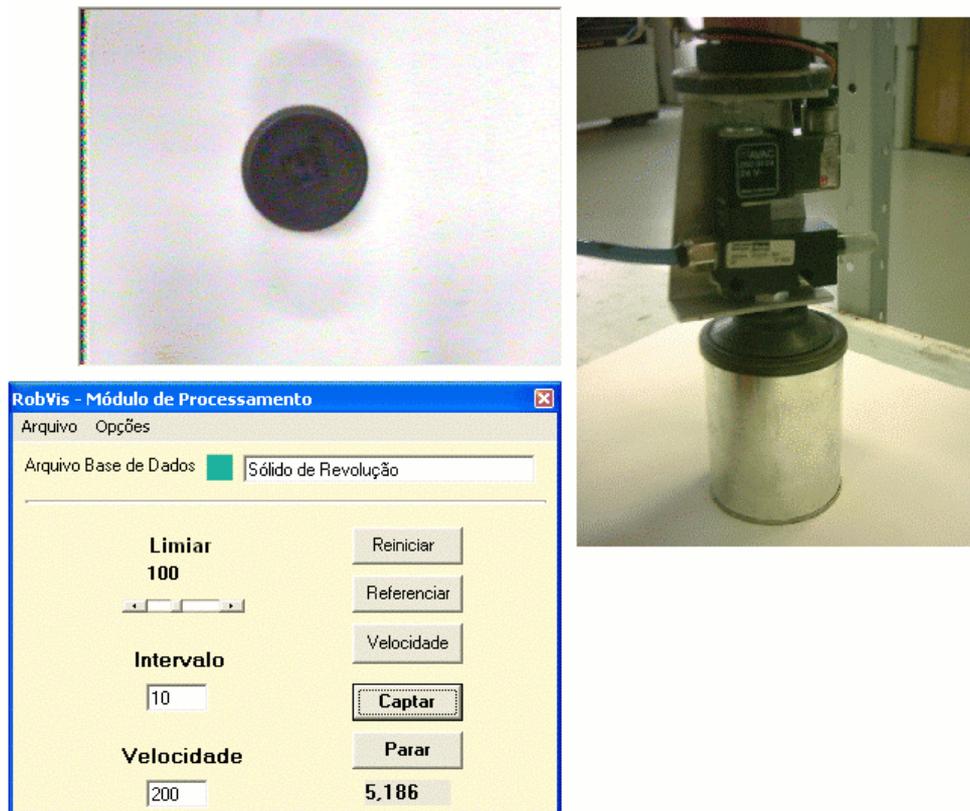


Figura 4.7 – Imagens de manipulação de sólido de revolução.

5. CONSIDERAÇÕES FINAIS

5.1 Resultados

Os resultados apresentados durante o período de avaliação foram altamente satisfatórios. O sistema se mostrou capaz de operar com os diferentes tipos de objetos utilizados. O desempenho alcançado para a combinação objeto escuro/fundo claro e objeto claro/fundo escuro foi idêntico e satisfatório. Os procedimentos de captura, processamento e análise da imagem são feitos de forma ágil pelo programa RobVis assim como a geração, transmissão e recepção de sinal. O programa *RobVisR.prg* rodando no controlador do robô executa a recepção dos comandos de forma rápida permitindo uma eficiente e precisa operação do manipulador. Estas considerações podem ser assumidas com base nos parâmetros adotados nos testes e na conseqüente rapidez requerida nas operações, da captação da imagem até a manipulação do objeto.

Os principais problemas determinados durante a avaliação, foram alguns erros de posicionamento da ferramenta para captura do objeto e poucos erros na determinação da orientação da ferramenta. Os erros de posicionamento são detectados quando a referência é feita de forma inadequada pelo operador e os erros de orientação são resultantes da análise imprecisa da imagem, visto que este procedimento é feito através de faixas de comparação em intervalos de 15°, o que pode acarretar algumas situações em que a ferramenta captura o objeto em uma faixa acima ou abaixo da ideal. Isto se deve à presença de algum ruído excepcional no momento da captura da imagem, distorcendo a representação do objeto e alterando o resultado dos cálculos. A Tabela 5.1 mostra os resultados obtidos, em três séries de 20 manipulações para as peças testadas, com destaque para os erros de orientação. Os erros de posicionamento não são analisados pois são inexistentes seguindo um correto processo de referenciamento.

Tabela 5.1 – Resultados dos testes.

Seção da Peça	Relação		Limiar	Ciclos de Manipulação	Erros de Orientação	Proporção de Erro (%)
	Objeto	Fundo				
"T"	Claro	Escuro	160	60	7	11,67
"T"	Escuro	Claro	100	60	5	8,33
Retangular	Claro	Escuro	160	60	6	10,00
Retangular	Escuro	Claro	100	60	5	8,33

5.2 Conclusões

O emprego de robôs está plenamente estabelecido em funções de automação industrial, mas, ainda assim, busca-se aumentar sua autonomia e capacidades com uma variada gama de sensores e dispositivos de tomada de decisão. Deste modo buscou-se, com este trabalho, desenvolver um sistema de visão dedicado a tarefas de manipulação visando ser integrado a um robô industrial. Deveria ser capaz de capturar imagens de uma porção do volume de trabalho do manipulador e de determinar a posição e a orientação de componentes a serem movimentados por ele, além de enviar esses dados ao seu controlador. Neste, outro componente do sistema estaria aguardando os comandos para coordenar a execução da tarefa.

Com base nos resultados obtidos nos testes de laboratório e tendo em vista os objetivos propostos quando do início deste trabalho, pode-se considerar a seguinte avaliação:

- O sistema de visão desenvolvido se mostrou satisfatório em tarefas de manipulação industrial tanto em condições de objeto escuro/fundo claro quanto objeto claro/fundo escuro e com necessidade ou não de se conhecer a orientação dos objetos a serem manipulados;
- Operacionalmente é de fácil implementação, visto que necessita de uma câmera CCD e placa de captura disponíveis no mercado, ligados a um PC onde roda o programa de captura/processamento/análise de imagens (RobVis), compatível com o ambiente *Windows*;
- Para a transmissão de sinais em tempo real utiliza-se um dispositivo (IRBCom) prático e de fácil execução, também desenvolvido no Laboratório de Robótica, como interface entre a porta paralela disponível em PC e placa E/S existente em controladores robóticos;
- O programa de manipulação RoBVisR apresenta igualmente desempenho satisfatório, embora customizado para robôs ABB, visto que utiliza linguagem Rapid nativa desses manipuladores. Para aplicação em outros equipamentos, deve apenas ser reescrito na linguagem correspondente.

5.3 Sugestões

Como sugestões para melhorias em trabalhos futuros nesta área, pode-se considerar:

- Substituição do procedimento de geração de tabela de comparação para busca da orientação de componentes através de um método indireto por um procedimento matemático que encontre diretamente a orientação do objeto na imagem captada.

- Implementação de métodos mais eficientes de referência para sanar eventuais desvios de posicionamento do manipulador durante o ciclo de operação decorrentes de imprecisão do método de referenciamento empregado.

- Ajuste automático do limiar com base em análise do histograma de tons de cinza da imagem;

- Testes com uma variedade maior de objetos com outras morfologias, para se estabelecer novas potencialidades e limitações do sistema obtido.

REFERÊNCIAS BIBLIOGRÁFICAS

- Azuela, J. H. S., 2001. “**Notes for the lecture: Invariant Moments Recognition**”, Instituto Politecnico Nacional, Mexico.
- Bart, E., Zetsche, C., Krieger, G., 1998, “Curvature Measures in Visual Information Processing”, **Open Systems and Information Dynamics**, vol. 5, pp. 25-39.
- Bayer, F. M., 2004. “Interface de Comunicação Paralela de Baixo Custo para Robô Industrial”, **Dissertação de Mestrado**, Programa de Pós-Graduação em Engenharia Mecânica, UFRGS.
- Bedworth, D. D., Henderson, M. R., Wolfe, P. M., 1991. “**Computer-Integrated Design and Manufacturing**”, McGraw-Hill, New York.
- Blasco, J., Aleixos, N., Moltó, E., 2003. “Machine Vision System for Automatic Quality Grading of Fruit”, **Biosystems Engineering**, vol. 85, pp. 415-423.
- Bonney, M. C., Yong, Y. F., editores, 1985. “**Robot Safety**”, IFS Publications Ltd., United Kingdom.
- Boucher, T.O., 1996, “**Computer Automation in Manufacturing**”, Chapman & Hall, London.
- Buchanan, W. J., Mahalik, N.P., 2003, “Some Studies on CAN Specification”, **Fieldbus Technology**, cap. 23, Springer, Berlin.
- Camarillo, D. B., Krummel, T. M., Salisbury, Jr., J. K., 2004. “Robotic Technology in Surgery: Past, Present and Future”, **The American Journal of Surgery**, vol. 188, pp. 2-15.
- Castleman, K. R., 1996. “**Digital Image Processing**”, Prentice Hall, Upper Sadle River
- Chen, S. B., Zhang, Y., Qiu, T. Lin, T., 2003. “Robotic Welding System with Vision-Sensing and Self Learning Neuron Control of Arc Welding Dynamic Process”, **Journal of Intelligence and Robotic Systems**, vol. 36, pp. 191-208.
- Chojnacki, W., Brooks, M. J., van den Engel, A., Gawley, D., 2004. “A New Constrained Parameter Estimator for Computer Vision Applications”, **Image and Vision Computing**, vol. 22, pp. 85-91.
- Colon, E., Hong, P., Habumuremyi, J., Doroftei, I., Baudoin, Y., Shali, H., Milojevic, D., Weemaels, J., 2002. “An Integrated Robotic System for Antipersonel Mines Detection”, **Control Engineering Practice**, vol. 10, pp. 1283-1291.
- Dong, Y., 2004, “Fast Computation of Variant Templates for Parallel Image Processing”, **Journal of Mathematical Imaging and Vision**, vol. 20, pp. 223-235.
- Fu, K. S., Gonzalez, R. C., Lee, C. S. G., 1987. “**Robotics: Control, Sensing, Vision and Intelligence**”, McGraw-Hill, New York.

- Gonzalez, R. C., Wintz, P., 1977. **“Digital Image Processing”**, Addison-Wesley, Massachussets.
- Gonzalez, R. C., Woods, R. E., 2003. **“Processamento de Imagens Digitais”**, Editora Edgard Blücher, São Paulo.
- González-Galván, E. J., Cruz-Ramirez, S. R., Seelinger, M. J., Cervantes-Sánchez, J. J., 2003. “An Efficient Multi-Camera, Multi-Target Scheme for the Three-Dimensional Control of Robots Using Uncalibrated Vision”, **Robotics and Computer Integrated Manufacturing**, vol. 19, pp. 387-400.
- Goodman, J. W., 1968. **“Introduction to Fourier Optics”**, McGraw-Hill, New York.
- Kragić, D., Petersson, L., Christensen, H. I., 2002. “Visually Guided Manipulation Tasks”, **Robotics and Autonomous Systems**, vol. 40, 193-203.
- Mahalik, N.P., 2003, “Industrial Network Standarts for Real-Time Distributed Control”, **Fieldbus Technology**, Springer, Berlin.
- Maia, P., 2004. “A Força de Submarinos Hoje”, **Tecnologia e Defesa – Edição Especial**, vol. 10, pp. 12-14.
- Maru, N., Kase, H., Yamada, S., Nishikawa, A., Miyazaki, F., 1993, “Manipulator Control by Visual Servoing With theStereo Vision”, **Proceedings of the 1993 IEEE/RSJ International Conference of Intelligent Robots and Systems**, Yokohama, Japan, pp. 1866-1870
- Murugesan, R., Mahalik, N. P., 2003, “Fieldbus and Contemporary Standarts”, **Fieldbus Technology**, cap. 1, Springer, Berlin.
- NASA, 1999. “A 640 x 486 Long Wavelength Infrared Câmera”, **NASA Tech Briefs**, June 1999, pp. 44-47.
- Niku, S. B., 2001. **“Introduction to Robotics: Analysis, Systems, Applications”**, Prentice Hall, Upper Saddle River.
- Nunes, J. C., Bouaoune, Y., Delechelle, E., Niang, O., Bunel, Ph., 2003. “Image Analisys by Bidimensional Empirical Mode Decomposition”, **Image and Vision Computing**, vol. 21, pp. 1019-1026.
- Okigawa, M., 1992. “A New and Simple Test Structure for Evaluating the Sectional Photo-Sensitivity Distribution of *Pixels* in a Frame-Transfer CCD Image Sensor”, **Proceedings of IEEE 1992 International Conference of Microeletronic Test Structures**, vol. 5, pp. 134-138.
- Pollefeys, M., 2004, “Visual Modeling with a Hand-Held Camera”, **International Journal of Computer Vision**, vol. 59(3), pp. 207-232.

- Ringer, M. Lasemby, J., 2004, "A Procedure for Automatically Estimating Model Parameters in Optical Motion Capture", **Image and Vision Computing**, vol. 22, pp. 843-850.
- Sallaberger, C., 1997. "Canadian Space Robotic Activities", **Acta Astronautica**, vol.41, pp. 239-246.
- Seitz, M., 1999. "Toward Autonomous Robotic Servicing: Using an Integrated Hand-Arm-Eye system for Manipulating Unknown Objects", **Robotics and Autonomous Systems**, vol. 26, pp. 23-42.
- Shi, H., 1997. "Two-Image Template operations for Binary Image Processing", **Journal of Mathematical Imaging and Vision**, vol. 3, pp. 269-274.
- Shirinzadeh, B., Tie, Y., 1993, "Vision-Based calibration of Robot Workstation for Off-Line Programming Systems", **Proceedings of an international Conference of the Australian Robot Association and the international federation of Robotics 1993**, pp. 531-538.
- Soares, A. M. S., 2004. "Controle de um Manipulador Robótico em uma Tarefa de *Pick and Place* Auxiliado por um Sistema de Visão", **Anais do III Congresso Nacional de Engenharia Mecânica**, Belém-PA, Brasil.
- Tunstel, E., Howard, A., 2003. "Approximate Reasoning for Safety and Survivability of Planetary Rovers", **Fuzzy Sets and Systems**, vol. 134, pp.27-46.
- Wilson, A., 1984. "Solid-State Camera Design and Application", **Machine Design**, April 1984, pp. 38-46.
- BARA, 17/12/2004 www.bara.org.uk/encyclopedia/robots/robintro.doc
- LACTEC, 12/11/2004 www.lactec.org.br
- IFTOMM, 17/12/2004 www.caip.rutgers.edu/iftomm
- RIA, 17/12/2004 www.roboticonline.com