

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

**SISTEMA DE VISÃO ROBÓTICA PARA RECONHECIMENTO DE CONTORNOS DE
COMPONENTES NA APLICAÇÃO DE PROCESSOS INDUSTRIAIS**

por

RENAN LUÍS FORESTI

Dissertação para obtenção do Título de
Mestre em Engenharia

Porto Alegre, dezembro de 2006

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

**SISTEMA DE VISÃO ROBÓTICA PARA RECONHECIMENTO DE CONTORNOS DE
COMPONENTES NA APLICAÇÃO DE PROCESSOS INDUSTRIAIS**

por

Renan Luís Foresti
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

Prof. Dr. Eduardo André Perondi – PROMEC/UFRGS

Prof. Dr. Ney Francisco Ferreira – DEMEC/UFRGS

Prof. Dr. Renato Ventura Bayan Henriques – DELET/UFRGS

Prof. Dr. Flávio José Lorini
Coordenador do PROMEC

Porto Alegre, dezembro de 2006.

RESUMO

O presente trabalho trata da implementação de um sistema de visão robótica para reconhecimento de formas bidimensionais e transformação do contorno em trajetória para um manipulador industrial. A aquisição da imagem ocorre através de uma câmera CCD sobre a área específica de captura. O uso de uma webcam também é testado. A imagem captada é enviada para o computador com processamento realizado em MATLAB, através de rotina de software de controle, escrita em VB.NET. São analisadas variações de contraste e resolução com objetos distintos, onde o sistema identifica os pixels que delimitam o contorno do objeto utilizando limiarização pelo Método de Otsu e algoritmos morfológicos. A posição de cada pixel é processada, transformada em coordenadas cartesianas e enviada para o controle do manipulador robótico, que efetua a trajetória, simulando um processo industrial. A transmissão ao controle do manipulador é realizada em protocolo especial, via porta paralela de um microcomputador à placa de aquisição de sinais digitais do controle do manipulador. Um processo de simulação em uma célula de manufatura proposto para validar o sistema, identifica objetos distintos que chegam de forma desordenada através de uma esteira transportadora.

Palavras-Chave: Sistemas de visão, processamento digital de imagens, robótica industrial, automação de processos.

ABSTRACT

This work approaches the implementation of a robotics vision system to recognize 2D forms and contour transformation in trajectory to an industrial manipulator. The image acquisition occur by CCD video camera on the specific capture area. A webcam system is also tested. The captured image is sent to a MATLAB computer processing, through a control software routine, written in VB. NET. The contrast and resolution changes are analyzed with different objects where the system identifies the pixels of object contour using Otsu's Thresholding Method and morphological algorithms. The position of each pixel is processed, transformed in cartesian coordinates and sent to the robotic manipulator control, which executes the trajectory simulating an industrial process. The transmission to the manipulator control is realized in a special protocol, using the parallel port of computer and a digital signal acquisition card of the manipulator control. A simulation process in a manufacturing cell is aimed to validate the system, identifying distinct objects that coming in a disorientated form from a belt conveyor.

Keywords: *Vision Systems, Digital Image Processing, Industrial Robotics, Automation Process*

ÍNDICE

	Pág	
1	INTRODUÇÃO.....	1
1.1	Descrição do Problema.....	2
1.2	Objetivo do Trabalho.....	2
2	REVISÃO BIBLIOGRÁFICA.....	4
2.1	Robô.....	4
2.2	Sistema de Visão Robótico.....	5
2.2.1	Diferenças entre o Sistema de Visão Humano e Artificial.....	5
2.2.2	Justificativas do uso de um Sistema de Visão Artificial.....	6
2.3	Aquisição da Imagem.....	7
2.4	Digitalização da Imagem Monocromática.....	8
2.4.1	Modelo Matemático de Imagem.....	8
2.4.2	Imagem Digital Monocromática.....	9
2.4.3	Imagem Digital Colorida.....	10
2.4.4	Amostragem e Quantização.....	10
2.5	Relacionamentos Básicos entre <i>Pixels</i>	11
2.5.1	Relação entre <i>Pixels</i> Adjacentes.....	11
2.5.2	Conectividade.....	12
2.5.3	Medidas de distância.....	13
2.5.4	Operações Lógico-Aritméticas.....	14
2.6	Pré-Processamento.....	14
2.6.1	Histograma.....	15
2.6.2	Realce.....	16
2.7	Segmentação.....	17
2.7.1	Limiarização (<i>Thresholding</i>).....	17
2.7.1.1	Limiarização Global Simples.....	19
2.7.1.2	Limiarização Ótima.....	19
2.7.1.3	Método de Otsu.....	21
2.8	Influência da Iluminação.....	22
2.8.1	Técnicas de Iluminação.....	23
2.9	Morfologia Matemática.....	24
2.9.1	Erosão.....	24
2.9.2	Dilatação.....	25

2.9.3	Abertura e Fechamento.....	26
2.9.4	Extração de Fronteiras.....	26
3	DESENVOLVENDO O SISTEMA.....	29
3.1	Equipamentos e Programas utilizados.....	29
3.1.1	Câmeras.....	29
3.1.2	Computador.....	29
3.1.3	Placa de Aquisição de Imagem.....	29
3.1.4	<i>Software MATLAB</i>	29
3.1.5	Dispositivo de comunicação - <i>Hardware IRBCom</i>	31
3.1.6	Robô modelo <i>ABB IRB1400</i>	32
3.1.7	<i>Software Boundary Lines</i>	33
3.2	Descrição e utilização do Aplicativo <i>Boundary Lines</i>	34
3.2.1	Identificação do Fundo.....	38
3.3	Validação Experimental do Sistema.....	39
3.3.1	Objetivo dos Testes.....	39
3.3.2	Bancada de Testes.....	39
3.3.3	Objetos Testados.....	40
3.3.4	Calibração do Sistema.....	40
4	RESULTADOS.....	41
4.1	Teste de variação de contraste e posição.....	41
4.2	Teste de envio de trajetórias.....	44
5	CONCLUSÕES.....	47
6	REFERÊNCIAS BIBLIOGRÁFICAS.....	48
	ANEXO I.....	51
	APÊNDICE A.....	52

LISTA DE SÍMBOLOS

CCD – *Charge Coupled Device* – Dispositivo de carga acoplada

C_0 – Grupo de pixels formados pelos valores $\{0, 1, 2, \dots, T-1\}$

C_1 – Grupo de pixels formados pelos valores $\{T, T + 1, \dots, L - 1\}$

dll – *Dinamic Library Link*

D_e – Distância Euclidiana

D_4 – Distância *City-block*

D_8 – Distância *tabuleiro de xadrez*

$E_1(T)$ – Probabilidade de considerar um ponto do objeto como sendo ponto de fundo

$E_2(T)$ – Probabilidade de considerar um ponto de fundo como sendo ponto do objeto

$f(x, y)$ – Função de intensidade luminosa

φ – Constante dependente da superfície

$g(x, y)$ – Função de intensidade luminosa transformada

$h(x, y)$ – Função normalizada de luminosidade

HSI – Padrão de cores (“matriz, saturação e intensidade”)

I – Intensidade de luz monocromática

i – Iluminação

IRBcom – Hardware de comunicação

$i_{máx}$ – Valor máximo de iluminação

$i_{mín}$ – Valor mínimo de iluminação

l – Intesidade do nível de cinza de uma imagem

L – Número de níveis de cinza

L – Valor máximo de nível de cinza

$L_{máx}$ – Valor máximo de nível de cinza

$L_{mín}$ – Valor mínimo de nível de cinza

$N_D(p)$ – Vizinhos diagonais do *pixel* p

$N_4(p)$ – 4-vizinhança do *pixel* p

$N_8(p)$ – 8-vizinhança do *pixel* p

P_1 – Probabilidade de nível de cinza

P_2 – Probabilidade de nível de cinza

$p(z)$ – Probabilidade de brilho Global

$p_1(z)$ – Probabilidade de brilho do objeto

$p_2(z)$ – Probabilidade de brilho do fundo

$p_r(r_k)$ = Percentual do k -ésimo nível de cinza
 R – Função de Transformação de níveis de cinza
 r – Reflectância
RAPID – Linguagem de programação dos robôs ABB
RGB – Padrão de cores (“*red, green, blue*”),
 $r_{máx}$ – Valor máximo de reflectância
 $r_{mín}$ – Valor mínimo de reflectância
 s_k – Função de Transformação
 σ_B^2 - Variância do Método de Otsu
 σ_1 – Desvio padrão
 σ_2 – Desvio padrão
 T - Limiar
USB – *Universal Serial Bus*
 μ_T – Média total do histograma
 μ_0 – Média do grupo C_0
 μ_1 – Média dos valores dos níveis de cinza
 μ_1 – Média do grupo C_1
 μ_2 – Média dos valores dos níveis de cinza
 V - conjunto dos valores de níveis de cinza conectados
 w_0 – Probabilidade do grupo C_0
 w_1 – Probabilidade do grupo C_1
YIQ – Padrão de cores (“luminância, em-fase, quadratura”)

ÍNDICE DE FIGURAS

Fig	TÍTULO	Pág
1.1	Etapas fundamentais no processamento de imagens.....	1
2.1	Tipos de robôs industriais.....	5
2.2	Convenção dos eixos para representação de imagens digitais.....	9
2.3	Cubos de cores RGB.....	10
2.4	Operações Lógico-Aritméticas (a) Região a, (b) Região b, (c) E, (d) OU, (e) NÃO e (f) XOU.....	14
2.5	Exemplos de histogramas.....	15
2.6	Exemplos da função de transformação de níveis de cinza.....	17
2.7	Histograma com um (a) limiar único, (b) com limiares múltiplos e (c) o resultado da limiarização.....	18
2.8	Técnicas de Iluminação. (a) Direta, (b) Difusa, (c) Traseira, (d) Estruturada e (e) Direcional.....	23
2.9	Erosão de A por B	25
2.10	Dilatação e A por B	25
2.11	O conjunto de <i>pixels</i> A e o Elemento Estruturante B	26
2.12	Início do processo de erosão.....	27
2.13	O processo erosão e <i>pixels</i> rotulados.....	27
2.14	Resultado do processo e erosão (a) e o contorno obtido(b).....	27
2.15	Elemento Estruturante.....	28
2.16	Resultado do processo de erosão (a) e o contorno obtido (b).....	28
3.1	Câmera CCD.....	29
3.2	Convenções utilizadas pela literatura (a) e (c), pelo <i>MATLAB</i> (b) e (d).....	30
3.3	<i>IRBCom</i> , painel frontal (a) conectores externos (b) componentes internos (c).....	31
3.4	Fluxograma de informações.....	32
3.5	Robô modelo - <i>ABB IRB1400</i>	32
3.6	Interface do <i>Boundary Lines</i>	33
3.7	Fluxograma das rotinas principais no <i>Boundary Lines</i>	34
3.8	Controle Principal do <i>Boundary Lines</i>	35
3.9	Interface do Controle Programa <i>RAPID</i> e <i>IRBCom</i>	35
3.10	Possibilidade de alteração de parâmetros.....	36
3.11	Blocos da <i>Simulação – Simulink</i>	36

3.12	Sucesso no processo de segmentação (a) <i>Simulink</i> e (b) <i>MATLAB</i>	37
3.13	Falha no processo de segmentação (a) <i>Simulink</i> e (b) <i>MATLAB</i>	37
3.14	Área de Captura, Reconhecimento e de Teste de Fundo.....	38
3.15	Exemplo de identificação de fundo.....	38
3.16	Bancada de Testes.....	39
3.17	Orientação dos eixos do plano de trabalho do robô (a), do plano da imagem (b) e eixos interpretados pelo <i>Boundary Lines</i> (c).....	39
3.18	Objetos e fundos testados.....	40
4.1	Objetos e seus respectivos histogramas respectivamente (a) e (d), (b) e (e), (c) e (f). Segmentação. Imagem binária com o fundo (g) branco, (h) pardo e (c) preto. O contorno obtido com o fundo (j) branco, (k) pardo e (l) preto. Figuras ampliadas (m), (n) e (o).....	42
4.2	Objetos e seus respectivos histogramas respectivamente (a) e (d), (b) e (e), (c) e (f). Segmentação. Imagem binária com o fundo (g) branco, (h) pardo e (c) preto. O contorno obtido com o fundo (j) branco, (k) pardo e (l) preto. Figuras ampliadas (m), (n) e (o).....	43
4.3	Trajetória obtida através da (a) câmera <i>CCD</i> e da (b) <i>webcam</i>	45
4.4	Trajetória obtida da câmera <i>CCD</i> (a) e comparação da trajetória com o objeto (b).	45
4.5	Trajetória obtida da <i>webcam</i> (a) e comparação da trajetória com o objeto (b).....	45
4.6	Trajetória obtida da <i>webcam</i> (a, b, c) e comparação da trajetória com o objeto (d).....	46

ÍNDICE DE TABELAS

Tab.	TÍTULO	Pág
2.1	Diferenças entre o Sistema Visual Humano e Artificial.....	6
2.2a	Número de <i>bits</i> (m) de armazenamento para imagens $N \times N$ ($N=M$).....	11
2.2b	Número de <i>bytes</i> (m) de 8 <i>bits</i> de armazenamento para imagens $N \times N$ ($N=M$).....	11

1 – INTRODUÇÃO

O surgimento do robô na indústria foi justificado principalmente pela necessidade de automação do processo produtivo, confiabilidade nos serviços executados e pela substituição do homem em locais de alta periculosidade. Segundo a definição do *Robot Institute of America*, “Um robô é um manipulador reprogramável e multifuncional, projetado para manipular materiais, peças, ferramentas ou dispositivos especializados, através de movimentos variáveis programados para o desempenho de uma variedade de tarefas”, porém, é difícil imaginarmos um manipulador reprogramável e multifuncional trabalhando isoladamente nos dias atuais. Com o avanço das áreas tecnológicas, fica evidente que o objetivo é a integração das máquinas, visando reduzir tempos de produção e conseqüentemente, custos. Os robôs ganham então, “sentidos”. Além da percepção sensorial, passam a se comunicar com outras máquinas, o que lhes garante uma inteligência artificial, podendo controlar e tomar decisões sobre o processo, sem a necessidade de intervenção humana. Um dos mais complexos desses “sentidos” e talvez o mais importante, é a visão, devido à riqueza de informações que uma imagem pode conter. Tenta-se então, “ensinar os robôs a enxergar”, criando um sistema de visão artificial, semelhante à dos seres humanos. Isto não é tão simples, embora o sistema de visão artificial consiga operar em praticamente todo o espectro de radiações eletromagnéticas e com altíssima velocidade de processamento, não possui a capacidade de trabalhar sob condições muito variadas (iluminação, realce, mudança de características, etc.). É geralmente inflexível, apresentando bom desempenho apenas nas condições em que foi projetado. Para tal propósito, é necessário conhecimento do procedimento utilizado no processamento digital de imagens, conforme a Figura 1.1.

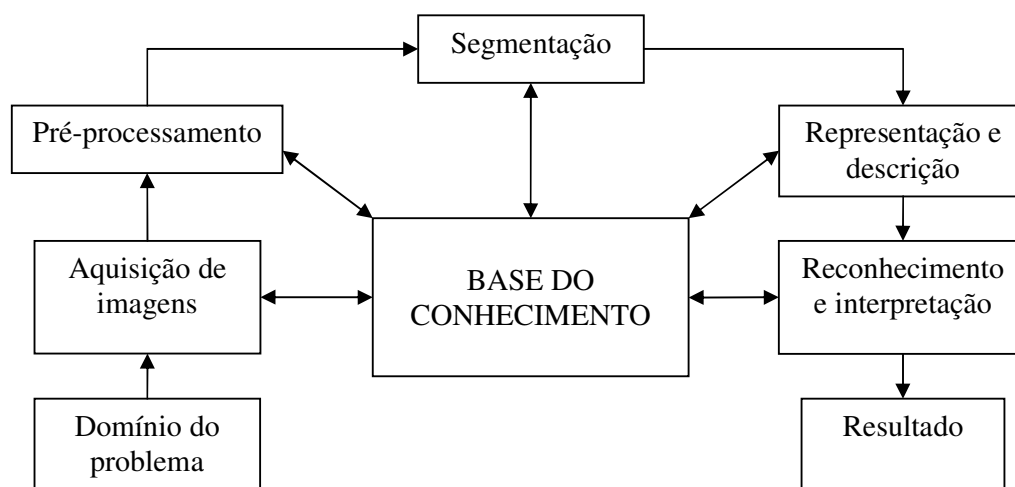


Figura 1.1 – Etapas fundamentais em processamento de imagens.

O primeiro passo no processamento digital de imagens, conforme Figura 1.1, é o *domínio do problema*, que consiste em definir qual informação tem que ser subtraída da imagem digital e de que forma esta informação estará disponível no resultado. Obter e digitalizar uma imagem fazem parte da *aquisição de imagens*. Dois elementos são necessários nesta etapa: um *sensor de imageamento*, isto é, um dispositivo físico que seja sensível a uma banda do espectro de energia e que produza um sinal elétrico proporcional a um nível de energia percebida, e um *digitalizador*, dispositivo para conversão do sinal elétrico do *sensor de imageamento* para a forma digital. O objetivo do *pré-processamento* é de melhorar a imagem de forma a se obter sucesso nos processos seguintes. Nesta etapa, são aplicadas técnicas de realce de contraste, remoção de ruído e isolamento de regiões de interesse. A etapa crítica do processamento de imagens é a *segmentação*, pois é nessa etapa que a imagem é dividida pelos seus objetos constituintes. Em geral, procura-se separar o objeto de interesse do fundo da imagem. Processos automáticos e não subjetivos de segmentação, que levam em conta as características da imagem, geralmente favorecem o sucesso do processamento. A escolha de uma *representação* é apenas parte da solução para transformar os dados provenientes da segmentação numa forma adequada para o processamento digital subsequente. O processo de *descrição* procura extrair características que sejam básicas para discriminação entre classes de objetos. O *reconhecimento* atribui um rótulo a um objeto, baseado na informação do seu descritor e a *interpretação*, envolve a atribuição de um significado a um conjunto de objetos reconhecidos. Recorre-se à *base do conhecimento* quando se dispõe de conhecimento prévio do resultado esperado. Além de guiar a operação de cada etapa no processamento, a base do conhecimento também controla sua interação (Gonzales; Woods, 2000).

1.1 – Descrição do Problema

A limitação imposta pela maioria dos processos de automação nas indústrias é de que nada seja variável. Os objetos têm dimensões e posições bem definidas, favorecendo unicamente operações com padrões, onde o *setup* de operação entre dois padrões pode custar horas de produção. A solução seria um Sistema de Visão integrado ao processo de automação que se adapte às alterações.

1.2 – Objetivo do Trabalho

O objetivo deste trabalho é de desenvolver um sistema de visão robótica que obtenha o contorno de objetos em posicionamentos diferentes, observando os seguintes fatores:

- Excluir a reprogramação de trajetórias dos contornos bidimensionais dos objetos. Este sistema deve ser capaz de obter a trajetória do contorno de qualquer forma bidimensional opaca, posicionada em qualquer posição e orientação;
- Minimizar a intervenção humana;
- Reconhecimento automático de contraste entre objeto e o fundo;
- Garantir o funcionamento do sistema com variações no tom cinza do fundo da imagem;
- Utilizar imagens estáticas, isto é, o objeto parado em relação à câmera no momento da captura;
- Utilizar separadamente dois sensores de imageamento com resoluções diferentes;
- Transmitir as coordenadas diretamente para o robô através de hardware ligado à porta paralela do computador e possibilitar a atuação do manipulador de acordo com a execução do programa residente no seu controle.

2 – REVISÃO BIBLIOGRÁFICA

A realização do trabalho proposto implica, essencialmente, em um estudo das técnicas de captura, processamento e análise de imagens e sua interação com instruções de programação para um manipulador robótico. Os principais temas e técnicas pertinentes são tratados na revisão bibliográfica que embasa a realização da aplicação desenvolvida e implementada.

2.1 – Robô

A origem da palavra *robô*, que em russo significa “trabalho”, é atribuído ao escritor tcheco, Karel Capeck, em sua peça de teatro de 1923, *Rossum's Universal Robots* (Alves, 1988). Para definir o que é um *Robô Industrial*, citam-se definições internacionais correntes:

“Uma máquina reprogramável para realizar funções de manipulação, movimentação e usinagem em condições de interagir com o ambiente e dotada de certo grau de liberdade” (*IFTToMM – International Federation for the Theory of Machines and Mechanism.*).

“Um robô é um manipulador reprogramável e multifuncional, projetado para manipular material, peças, ferramentas ou dispositivos especializados, através de movimentos variáveis programados para o desempenho de uma variedade de tarefas” (*RIA – Robot Institute of America*).

“Um dispositivo programável que seja projetado para manipular e transportar peças, objetos e ferramentas” (*BRA – British Robot Association*).

Autonomia, versatilidade e reprogramação, são características que diferenciam os robôs de uma máquina automática projetada para uma determinada função não facilmente modificável (Lorini, 2003). Do ponto de vista macroscópico, um sistema robótico pode ser dividido em *mecanismo*, *acionamento* e *sistema de controle*. O *mecanismo* está relacionado com a parte mecânica móvel e subdivide-se em: *braço*, *punho* e *efetuador*. O *braço* tem a função de posicionar o efetuador, enquanto o *punho*, de orientar. O *efetuador* constitui a ferramenta e é escolhido em função da tarefa a ser executada. O *acionamento*, que também é parte física do *mecanismo*, é responsável pela movimentação de cada junta do *braço* e do *punho*. O *sistema de controle* é responsável pela coordenação e controle do movimento de todas as juntas (Alves, 1988). Em relação às juntas, elas podem ser *prismáticas* (P) ou *rotacionais* (R). *Juntas prismáticas* apresentam maior rigidez e precisão, porém são de construção complexa quando comparadas às *juntas rotacionais* (Lorini, 2003). A combinação das juntas principais do *braço* classifica um robô industrial em diferentes categorias construtivas, que por sua vez se identificam com linhas de aplicação específicas na automação de processos, de acordo com respectivos volumes de trabalho e desempenho operacional.

Os principais modelos de manipuladores industriais atuais enquadram-se nas seguintes categorias, ilustradas na figura 2.1:

- Cartesiano ou Retangulares, as juntas são todas prismáticas (PPP);
- Cilíndricos, uma junta rotacional e as demais prismáticas (RPP);
- Esféricos, duas juntas rotacionais e uma prismática (RRP);
- Antropomorfos ou articulados, todas juntas rotacionais (RRR);
- SCARA, duas juntas rotacionais e uma prismática (RRP).

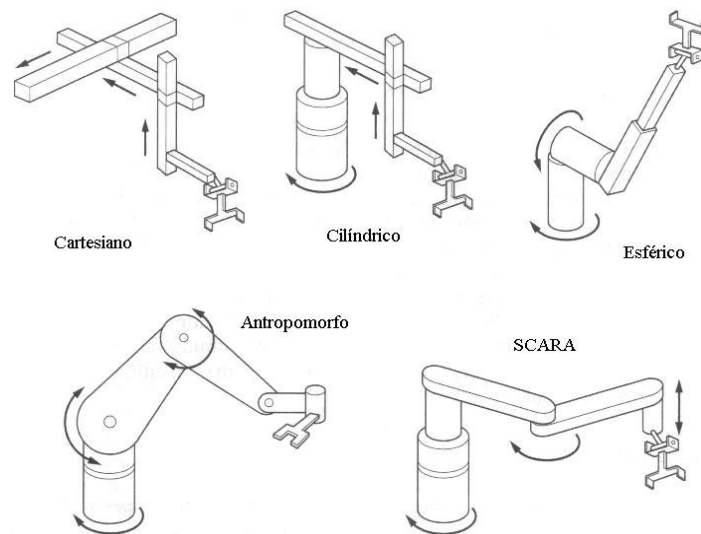


Figura 2.1 – Tipos de configurações para robôs industriais.

2.2 – Sistema de Visão Robótico

Robôs dotados de um sistema de visão possuem maior flexibilidade, autonomia e certa inteligência artificial, permitida através de captura, processamento e interpretação de imagens de objetos cujas posições e orientações passam a ser conhecidas via sistema computacional de apoio associado ao controle do manipulador. São capazes de interagir com o meio de trabalho de forma a responder em tempo real a alterações operacionais.

2.2.1 – Diferenças entre o Sistema de Visão Humano e Artificial

O grande desafio é fazer um sistema de visão artificial semelhante ao sistema de visão humano, que trabalhe em diferentes condições de luminosidade e contraste sem perder a capacidade de interpretar a cena. A Tabela 2.1 lista as principais diferenças entre os dois sistemas de visão. (Marques Filho; Vieira Neto, 1999).

Tabela 2.1 - Diferenças entre o Sistema Visual Humano e Artificial.

	Sistema Visual Humano	Sistema de Visão Artificial
Espectro	Limitado à faixa de luz visível (300nm a 700nm) do espectro de ondas eletromagnéticas.	Pode operar em praticamente todo o espectro de radiações eletromagnéticas, dos raios X ao infravermelho.
Flexibilidade	Extremamente flexível, capaz de se adaptar a diferentes tarefas e condições de trabalho.	Normalmente inflexível. Apresenta bom desempenho apenas na tarefa para qual foi projetado.
Habilidade	Pode estabelecer estimativas relativamente precisas em assuntos subjetivos.	Pode efetuar medições exatas, baseadas em contagem de pixels e, portanto, dependentes da imagem digitalizada.
Cor	Possui capacidade de interpretação subjetiva de cores.	Mede objetivamente os valores das componentes R, G e B para determinação de cor.
Sensibilidade	Capaz de se adaptar a diferentes condições de luminosidade, características físicas da superfície do objeto e distância do objeto. Limitado na distinção de muitos níveis diferentes de cinza, simultaneamente.	Sensível ao nível e padrão de iluminação, bem como à distinção em relação ao objeto e suas características físicas. Pode trabalhar com centenas de tons de cinza, conforme o projeto do digitalizador.
Tempo de resposta	Em torno de 0,1s	Dependente de aspectos de hardware, podendo ser da ordem de 0,001s.
2D e 3D	Pode executar tarefas 3D e com múltiplos comprimentos de onda (dentro do espectro visível) facilmente.	Executa tarefas 2D com relativa facilidade, mas é lento e limitado em tarefas 3D
Percepção	Percebe variações de brilho em escala logarítmica. A interpretação subjetiva de brilho depende da área ao redor do objeto considerado.	Pode perceber brilho em escala linear ou logarítmica.

2.2.2 – Justificativas do uso de um Sistema de Visão Artificial

Segundo (Alves, 1988), existem três gerações de robôs. A primeira geração, considerada de baixo grau de desenvolvimento tecnológico, não possuía sensores e sim paradas mecânicas. A segunda geração, considerada de médio grau de tecnologia, possuía sensores internos e, desta maneira, eram capazes de conhecer o seu estado em relação à sua base, porém ainda impossibilitados de reconhecer modificações imprevistas em seu ambiente de trabalho. A utilização de controle por malha fechada já era possível. A terceira geração (atual) é capaz de reconhecer modificações em seu ambiente de trabalho e de reagir de forma inteligente. Isto possibilitou maior flexibilidade dos processos produtivos com conseqüente aumento de produtividade. Uma subdivisão da terceira geração em relação ao nível de inteligência artificial se faz necessária. Os robôs de baixo nível de inteligência artificial são dotados de sensores que apenas identificam mudanças no ambiente de trabalho, porém não quantificam e nem qualificam estas mudanças. Pode-se citar o exemplo de uma esteira transportadora que alimenta a célula de trabalho de um robô, com sensor de presença de objetos. Neste caso, o objeto transportado pela esteira deve ter sua posição e orientação fixa, além de manter suas dimensões. O sensor externo, apenas enviaria um sinal de presença que seria interpretado pelo comando do robô. O controle ficaria reduzido a dois estados. Os robôs de alto nível de inteligência artificial são dotados de

sensores capazes de quantificar e qualificar as modificações no seu ambiente de trabalho, ou seja, os objetos manipulados agora podem ter posições e orientações diferentes, além disso, pode-se trabalhar com objetos de forma e dimensões desiguais. *Sensores de imageamento*, ligados a algoritmos de visão artificial, são responsáveis por tornar o robô e o processo produtivo mais flexíveis. Citam-se outras vantagens da utilização de um robô com sistema de visão artificial:

- Possibilidade de inspeção, aferição e alinhamento de peças;
- Manipulação de objetos sem posição pré-definida;
- Manipulação de objetos diferentes;
- Manipulação de objetos que possuem variação dimensional;
- Possibilidade de trabalhar com peças em movimento;
- Maior nível de inteligência artificial e maior controle do processo produtivo;
- Aumento de produtividade.

É evidente que um sistema robótico com sistema artificial de visão é mais caro que um robô com sensores de presença e, portanto, é o processo produtivo que define a necessidade da utilização de um robô com ou sem sistema artificial de visão.

2.3 – Aquisição da Imagem

Para tornar possível a análise computacional sobre imagens, é necessário que os dados estejam na forma digital, isto é, a luminosidade deve ser convertida em sinais elétricos distribuídos espacialmente. Uma imagem adquirida deve ser convertida em sinal elétrico analógico através de sensores ópticos, sendo posteriormente, convertida em sinal digital por circuitos eletrônicos chamados *frame grabbers* (dispositivo de captura de quadros). Os sensores de maior destaque são os sensores a válvula e os sensores de estado sólido.

O princípio dos sensores a válvula consiste na varredura de uma superfície fotossensível à base de fósforo por um feixe eletrônico. A intensidade de corrente do feixe eletrônico é proporcional à intensidade luminosa no ponto em questão. A câmera Vidicon é um exemplo de sensores a válvula, que apresentam a desvantagem de serem volumosos e difíceis de ajustar além do consumo de energia e aquecimento.

Um dos sensores de estado sólido mais utilizado atualmente para aquisição de imagens é o dispositivo de carga acoplada - CCD (*Charge Coupled Device*). A tecnologia das câmeras CCD baseia-se em pastilhas semicondutoras com determinado número de recipientes capazes de armazenar carga elétrica, proporcional à intensidade luminosa. A quantidade de carga elétrica armazenada nos recipientes corresponde a valores analógicos, o que significa dizer, que erros de

quantização são praticamente nulos. Desta maneira, as câmeras CCD são equivalentes a uma memória analógica, cujos dados são acessados serialmente através da transferência de carga entre os recipientes.

Os sensores de imagem normalmente fornecem sinais elétricos analógicos variantes no tempo, sendo que o *frame grabbers* é responsável pelo sincronismo horizontal e vertical do sinal composto de vídeo, determinando os limites de informação da imagem. A captura de um quadro inicia após o pulso de sincronismo vertical, onde o sinal analógico é amostrado no tempo, passando por uma conversão analógico/digital sendo armazenado em memória. Esta memória de armazenamento volátil é denominada *frame buffer*. O processo se repete até o próximo pulso de sincronismo vertical, ou seja, o final do quadro. “Os pulsos de sincronismos horizontal e vertical servem de referência para a montagem da matriz de dados da imagem, juntamente com as informações de quantidade de amostras por linha (resolução horizontal) e quantidade de linhas propriamente dita (resolução vertical)” (Marques Filho; Vieira Neto, 1999).

Sistemas de captura de imagens coloridas necessitam de três sensores monocromáticos operando em conjunto com filtro de cores. O sincronismo é comum aos três circuitos, porém, é necessário um sistema complementar para a decodificação do sinal nas três componentes de cor.

2.4 – Digitalização da Imagem

2.4.1 – Modelo Matemático de Imagem Monocromática

Define-se imagem como uma função discreta de intensidade luminosa $f(x, y)$, em que o valor de f em um ponto (x, y) qualquer, representa a intensidade (brilho) desse ponto. Sabe-se que a luz é energia, e por isso, $f(x, y)$ deve ser finita e positiva, ou seja:

$$0 < f(x, y) < \infty \quad (1)$$

A natureza de $f(x,y)$ pode ser caracterizada por dois componentes: a *iluminação* $i(x, y)$ e a *reflectância* $r(x,y)$, onde pode-se escrever que o produto de $i(x, y)$ por $r(x, y)$ resulta $f(x, y)$.

$$f(x,y) = i(x, y) r(x, y) \quad (2)$$

onde:

$$0 < i(x, y) < \infty \quad (3)$$

e

$$0 < r(x, y) < 1 \quad (4)$$

Isso significa que a reflectância é limitada entre a absorção total, $r(x, y) = 0$; e a reflectância total, $r(x, y) = 1$. No processamento de imagens digitais, tem-se que melhorar os limites da função $f(x, y)$, e para isso, considera-se “ l ” como a intensidade do nível de cinza de uma imagem monocromática em um determinado ponto. Desta maneira:

$$L_{min} \leq l \leq L_{máx} \quad (5)$$

onde a única restrição sobre L_{min} é que seja um valor positivo e sobre $L_{máx}$ é que seja finito. Na prática $L_{min} = i_{min} r_{min}$ e $L_{máx} = i_{máx} r_{máx}$. Desta maneira, tem-se uma *escala de cinza*, definida pelo intervalo $[L_{min}, L_{máx}]$. Por simplificação, costuma-se deslocar o intervalo para $[0, L]$, onde $l = 0$ é considerado negro e $l = L$ branco. Qualquer valor intermediário entre $[0, L]$ será um tom de cinza, obrigatoriamente para imagens monocromáticas (Gonzales; Woods, 2000).

2.4.2 – Imagem Digital Monocromática

Uma imagem digital pode ser representada por uma função discreta de intensidade de luz, $f(x, y)$, onde x e y representam as coordenadas espaciais e f o brilho proporcional em cada ponto (x, y) . Em imagens monocromáticas, f está relacionado com os *níveis de cinza* em cada ponto. Desta maneira, pode-se representar uma imagem digital por uma matriz, onde cada elemento corresponde a um ponto da imagem digital; e os valores de seus elementos, correspondem ao nível de cinza da imagem, conforme Figura 2.2 (Gonzales; Woods, 2000).

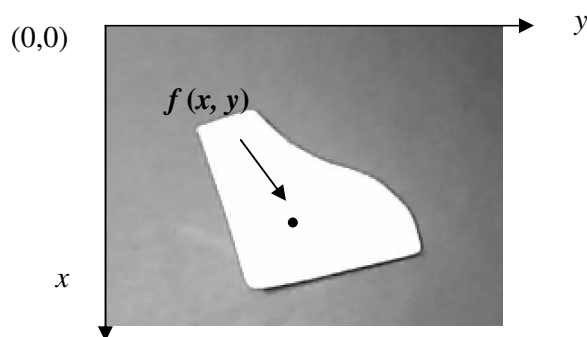


Figura 2.2 – Convenção dos eixos para representação de imagens digitais.

A matriz que representa uma imagem digital é denominada *matriz digital*, e seus elementos são denominados de “*pixels*”, também conhecidos como elementos da imagem, elementos da figura ou ainda “*pels*” (*picture elements*) (Marques Filho; Vieira Neto, 1999).

2.4.3 - Imagem Digital Colorida

Os modelos de cores mais utilizados no processamento de imagens são o RGB (“red, green, blue”), o YIQ (“luminância, em-fase, quadratura”) e HSI (“matriz, saturação e intensidade”). O modelo RGB é o mais utilizado por câmeras e monitores de vídeo, o modelo YIQ é usado na transmissão comercial de TV colorida (padrão NTSC) e o modelo HSI é mais utilizado para manipulação de imagens coloridas (Gonzales; Woods, 2000).

A conversão entre os modelos pode ser estabelecida através de formulações que consideram as devidas correspondências. O modelo RGB é bem representado por um cubo em um sistema de coordenadas cartesianas, onde três de seus vértices são as cores primárias (vermelho, verde e azul) e os outros três as cores secundárias (amarelo, ciano, magenta). O vértice mais junto à origem é o preto e o vértice mais afastado é o branco. A escala de cinza é representada pela diagonal do cubo que vai do vértice preto até o vértice branco. Por conveniência, assume-se o cubo – Figura 2.3 – normalizando, onde os valores de R, G e B variam de 0 a 1.

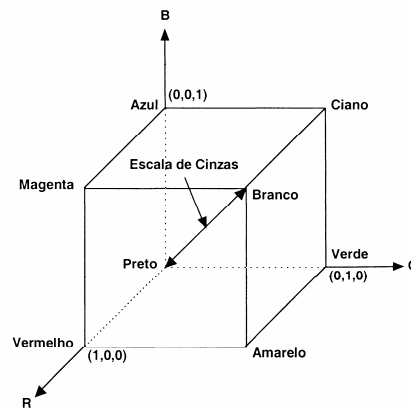


Figura 2.3 – Cubo de cores RGB.

Para o modelo RGB, o mais utilizado em câmeras de vídeo, em termos práticos, trabalha-se com uma representação para a intensidade monocromática (I) em proporções dos componentes de cor vermelha, verde e azul, definida pela seguinte formulação (Staugaard, 1987)

$$I = 0,3 \times R + 0,59 \times G + 0,11 \times B \quad (6)$$

2.4.4 – Amostragem e Quantização

A digitalização das coordenadas espaciais (x, y) é denominada *amostragem da imagem* e a digitalização da amplitude é chamada *quantização dos níveis de cinza*. A imagem contínua em $f(x, y)$ é aproximada por uma matriz $N \times M$, onde cada elemento representa um pixel. A matriz guarda a informação da posição e o nível de cinza de cada pixel.

$$f(x, y) \approx \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix} \quad (7)$$

A *resolução*, que é o grau de discernimento de detalhes, depende da amostragem e da quantização. Quanto maiores esses parâmetros, melhor será a aproximação da imagem original pela matriz digital. Porém, o inconveniente da alta resolução, tamanho de armazenamento e processamento computacional, é mostrado nas Tabelas 2.2a e 2.2b. Além do tamanho de armazenamento aumentar consideravelmente com o aumento de N e M , o processamento computacional também aumenta. A definição de uma “boa” imagem é uma qualidade subjetiva e que depende muito da aplicação, mas, sabe-se que a qualidade de uma imagem está relacionada não apenas com a amostragem ou com a quantização, e sim com esses dois parâmetros. Imagens onde se reduz o número de *pixels* (Ex: de 512x512 para 64x64), pode produzir um efeito de xadrez, e imagens onde se reduz a quantização (Ex: de 256 para 8 níveis de cinza) pode apresentar um efeito muito comum denominado de *falso contorno* (Gonzales; Woods, 2000).

Tabela 2.2a – Número de *bits* (m) de armazenamento para imagens $N \times N$ ($N=M$).

N	m = 1	m = 2	m = 3	m = 4	m = 5	m = 6	m = 7	m = 8
32	1.024	2.048	3.072	4.096	5.120	6.144	7.168	8.192
64	4.096	8.192	12.288	16.384	20.480	24.576	28.672	32.768
128	16.384	32.768	49.152	65.536	81.920	98.304	114.688	131.072
512	262.144	524.288	786.432	1.048.576	1.310.720	1.572.864	1.835.008	2.097.152
1.024	1.048.576	2.097.152	3.145.728	4.194.304	5.242.880	6.291.456	7.340.032	8.388.608

Tabela 2.2b – Número de *bytes* (m) de 8 *bits* de armazenamento para imagens $N \times N$ ($N=M$).

N	m = 1	m = 2	m = 3	m = 4	m = 5	m = 6	m = 7	m = 8
32	128	256	384	512	640	768	896	1.024
64	512	1.024	1.536	2.048	2.560	3.072	3.584	4.096
128	2.048	4.096	6.144	8.192	10.240	12.288	14.336	16.384
512	32.768	65.536	98.304	131.072	163.840	196.608	229.376	262.144
1.024	131.072	262.144	393.216	524.288	655.360	786.432	917.504	1.048.576

2.5 – Relacionamentos Básicos entre *Pixels*

2.5.1 – Relação entre *Pixels* Adjacentes

Para um pixel p com coordenadas (x, y) , pode-se afirmar que este *pixel* possui dois vizinhos horizontais, dois verticais e quatro diagonais. As coordenadas dos *pixels* adjacentes horizontais e verticais são respectivamente: $(x + 1, y)$, $(x - 1, y)$, $(x, y + 1)$, $(x, y - 1)$. Esse

conjunto de *pixel* é denominado de *4-vizinhança* de p , e é representado por $N_4(p)$. Todos os *pixels* estão distantes uma unidade em relação ao *pixel* p , assim, quando o *pixel* p estiver na borda da imagem, um ou dois *pixels* adjacentes não estarão presentes. As coordenadas dos *pixels* diagonais são representadas por: $(x + 1, y + 1)$, $(x + 1, y - 1)$, $(x - 1, y + 1)$, $(x - 1, y - 1)$ e denotados por $N_D(p)$. Agrupando os *pixels* diagonais com os horizontais e verticais, tem-se um conjunto de *pixels* denominado *8-vizinhança* de p , representados por $N_8(p)$. Analogamente à *4-vizinhança*, alguns vizinhos do *pixel* p , não estarão presentes se este estiver nas bordas (Gonzales; Woods, 2000).

2.5.2 – Conectividade

Um conceito importante e muito utilizado para a determinação dos limites (contorno) e componentes de regiões de um objeto é o de *conectividade*. Dois *pixels* estão conectados se de alguma forma são adjacentes e seus níveis de cinza satisfazem um critério similar. Por exemplo, em uma imagem binária, dois *pixels* lado a lado estão conectados somente se possuírem o mesmo valor, 0 ou 1. Em imagens com 256 níveis de cinza, pode-se estabelecer um critério de conectividade para um intervalo de níveis de cinza, ou seja, dois *pixels* lado a lado estarão conectados se possuírem valores de níveis de cinza entre 130 e 140, por exemplo. Desta maneira tem-se o conjunto V dos valores de níveis de cinza utilizados para definir a conectividade, que no exemplo anterior seria entre 130 e 140, ou seja, $V = \{130, 131, \dots, 139, 140\}$.

Os tipos de conectividade que podem ser considerados correspondem a:

- a) *4-Conectividade*: Assumindo valores estabelecidos em V , dois *pixels* p e q estarão *4-conectados* se q pertencer ao conjunto $N_4(p)$.
- b) *8-Conectividade*: Assumindo valores estabelecidos em V , dois *pixels* p e q estarão *8-conectados* se q pertencer ao conjunto $N_8(p)$.
- c) *m-Conectividade*: Assumindo valores estabelecidos em V , dois *pixels* p e q estarão *m-conectados* se:
 - q está em $N_4(p)$, ou;
 - q está em $N_D(p)$ e a intersecção entre os conjuntos $N_4(p)$ e $N_4(q)$ for vazio.

A *m-conectividade*, também conhecida como *conectividade mista*, é uma modificação da *8-conectividade*. Sua finalidade é de eliminar as conexões por múltiplos caminhos, que sempre aparecem quando a *8-conectividade* é utilizada. Um *pixel* p é *adjacente* a um *pixel* q se eles forem conectados. O tipo de adjacência (4, 8 ou m) depende do tipo de conectividade que foi utilizado (Marques Filho; Vieira Neto, 1999).

2.5.3 – Medidas de distância

Sejam os pixels p , q e z , com coordenadas (x, y) , (s, t) e (u, v) respectivamente, e D a função distância que será válida se:

- $D(p, q) \geq 0$;
- $D(p, q) = D(q, p)$, e;
- $D(p, z) \leq D(p, q) + D(q, z)$.

Definem-se três tipos de distâncias, conforme definições e ilustrações correspondentes a seguir (Gonzales; Woods, 2000):

- **Distância Euclidiana:** É a distancia radial.

$$D_e(p, q) = [(x - s)^2 + (y - t)^2]^{1/2} \quad (8)$$

$\sqrt{12}$	$\sqrt{6}$	2	$\sqrt{6}$	$\sqrt{12}$
$\sqrt{6}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{6}$
2	1	0	1	2
$\sqrt{6}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{6}$
$\sqrt{12}$	$\sqrt{6}$	2	$\sqrt{6}$	$\sqrt{12}$

- **Distância D_4 (city-block):** Possui a forma de um losango.

$$D_4(p, q) = |x - s| + |y - t| \quad (9)$$

		2		
	2	1	2	
2	1	0	1	2
	2	1	2	
		2		

- **Distância D_8 (tabuleiro de xadrez).** Neste caso, usa-se a maior distância pelo cálculo de $|x - s|$ ou $|y - t|$.

$$D_8(p, q) = \max(|x - s|, |y - t|) \quad (10)$$

2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

2.5.4 – Operações Lógico-Aritméticas

Operações lógicas e aritméticas entre *pixels* são amplamente utilizadas na maioria dos processamentos digitais de imagens. Considerando dois *pixels* p e q , as operações aritméticas são mostradas a seguir (Marques Filho; Vieira Neto, 1999):

- Adição: $p + q$; usada para fazer a média de imagens para redução de ruído.
- Subtração: $p - q$; usada para remover informações estáticas de fundo.
- Multiplicação: pq ; usada para corrigir sombras.
- Divisão: p/q ; usada para corrigir sombras.

As operações aritméticas são feitas *pixel a pixel*.

Considere as regiões a e b , conforme as Figuras 2.4 (a) e 2.4 (b), respectivamente. As principais operações lógicas entre as regiões a e b são as seguintes:

- E: $a \cap b$, Figura 2.4 (c);
- OU: $a \cup b$, Figura 2.4 (d);
- NÃO: Complemento de a , Figura 2.4 (e);
- XOU: $a \cup b - a \cap b$, Figura 2.4 (f).

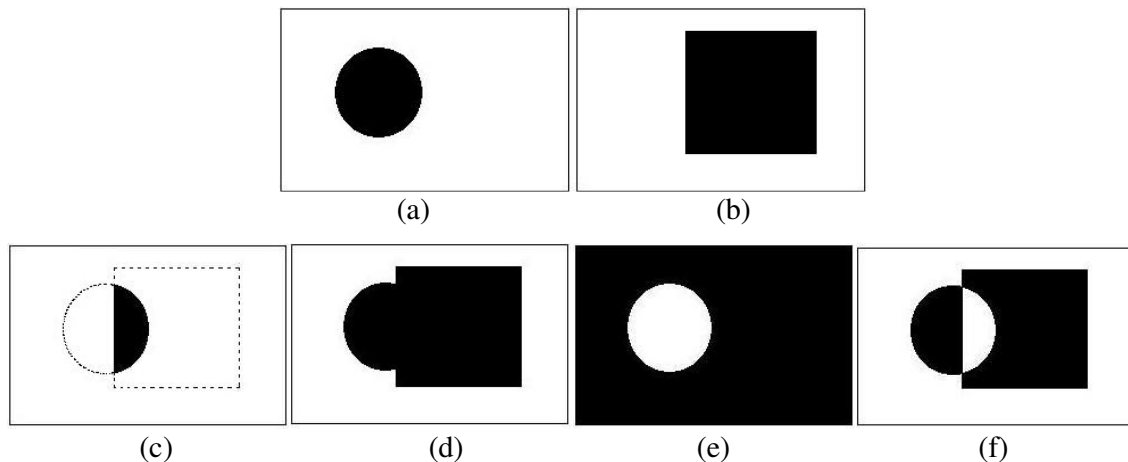


Figura 2.4 – Operações Lógico-Aritméticas (a) Região a, (b) Região b, (c) E, (d) OU, (e) NÃO e (f) XOU.

2.6 – Pré-Processamento

No tratamento de uma imagem, a etapa de pré-processamento objetiva melhorar a imagem para se obter sucesso nos processos seguintes, principalmente no processo de segmentação.

2.6.1 – Histograma

Partindo de uma imagem digital em níveis de cinza, pode-se representar a intensidade luminosa desta imagem em um gráfico, considerando no eixo das abscissas os níveis de cinza e no eixo das ordenadas, o número de pixels. Este gráfico é chamado de *histograma*. O *histograma*, ao invés de conter a informação do número de pixels em cada nível de cinza, geralmente contém o percentual de pixels em cada nível, facilitando sua interpretação. Através da visualização do *histograma* de uma imagem, tem-se uma indicação qualitativa quanto ao nível de contraste e brilho. É importante ressaltar que o *histograma* não tem nenhuma ligação com o posicionamento dos objetos na imagem. Duas imagens de mesmo tamanho (mesmo número de *pixels*) de um objeto claro em um fundo escuro, produzem o mesmo histograma para este objeto em posições diferentes, porém garantido que o número de *pixels* claros e escuros são os mesmos nas duas imagens. Cada termo do histograma é calculado conforme a seguinte equação.

$$p_r(r_k) = \frac{n_k}{n} \quad (11)$$

onde: $0 \leq r_k \leq 1$; $k = 0, 1, 2, \dots, L - 1$; L é o número de níveis de cinza da imagem; n é o número total de pixels na imagem; $p_r(r_k)$ = percentual do k -ésimo nível de cinza; n_k = número de pixels cujo nível de cinza corresponde a k . Os histogramas ilustrados a seguir, correspondem respectivamente, a uma imagem (a) escura, (b) clara, (c) de baixo contraste e (d) de alto contraste.

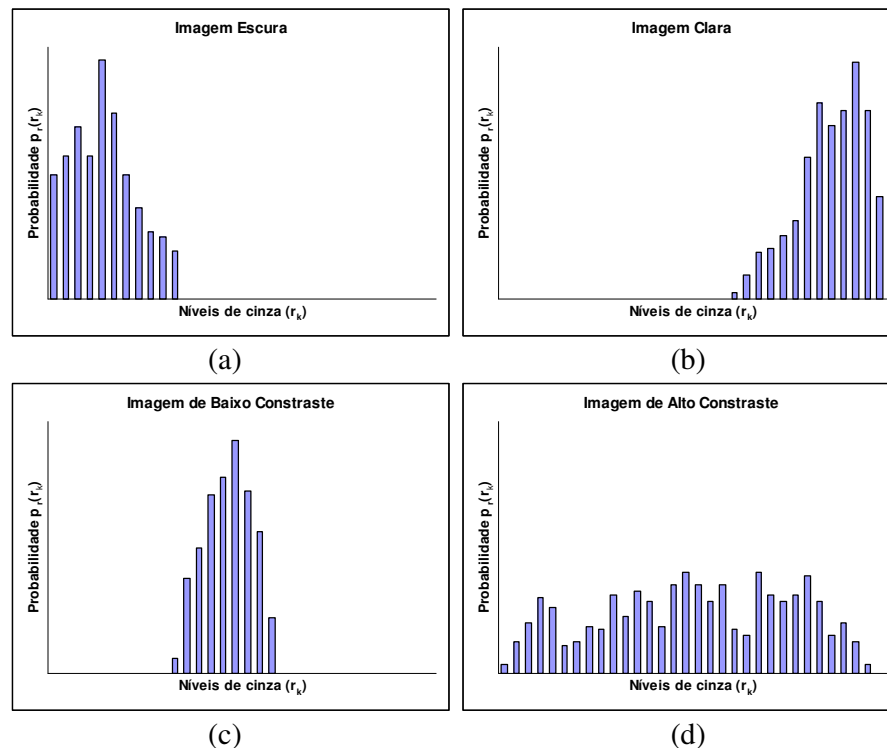


Figura 2.5 – Exemplos de histogramas.

O conceito de *histograma* também é aplicado às imagens coloridas. Neste caso a imagem é decomposta em seus componentes R (*red*), G (*green*), B (*blue*) e para cada componente é calculado o *histograma*.

2.6.2 – Realce

O principal objetivo das técnicas de realce é de modificar a imagem de maneira que seja mais útil para uma aplicação específica. As técnicas de realce são bastante dependentes da aplicação, o que significa dizer que não existe uma técnica ótima para todos os casos. As técnicas de realce podem ser subdivididas em duas categorias: *técnicas no domínio espacial* e *técnicas no domínio frequência*. As *técnicas no domínio espacial* propõem a manipulação direta dos pixels da imagem, enquanto que as *técnicas no domínio frequência*, propõem modificações da Transformada de Fourier da imagem (Gonzales; Woods, 2000). A seguir, são listadas as principais técnicas no domínio espacial, que também podem ser utilizadas no domínio frequência.

Equalização

Equalizar um histograma significa redistribuir os valores de tons de cinza, a fim de obter-se um histograma uniforme. Para isto, utiliza-se uma função auxiliar, denominada de *função de transformação*. Utilizar a *função de distribuição acumulada* (*cdf – cumulative distribution function*), Equação (12), é a forma mais usual para se equalizar um histograma. (Marques Filho; Vieira Neto, 1999).

$$s_k = \sum_{j=0}^k \frac{n_j}{n} = \sum_{j=0}^k p_r(r_j) \quad (12)$$

onde: $0 \leq r_j \leq 1$; $k = 0, 1, 2, \dots, L - 1$, L é o número de níveis de cinza da imagem.

Processamento ponto-a-ponto

Neste caso, o realce depende apenas do nível de cinza do ponto em questão. Pode-se expressar as funções de processamento ponto-a-ponto na forma da equação:

$$g(x, y) = Rf(x, y) \quad (13)$$

onde: $f(x, y)$ é a imagem de entrada; $g(x, y)$ é a imagem processada; R é a função de transformação de níveis de cinza (função mapeamento).

A função R pode ser linear ou não-linear, porém deve satisfazer duas condições:

- I. A função R deve retornar um único valor, $g(x, y)$, para cada valor distinto de $f(x, y)$.

$$\text{II. } 0 \leq Rf(x, y) \leq 1 \text{ para } 0 \leq f(x, y) \leq 1$$

Desta maneira, a função R mapeará cada pixel da imagem original $f(x, y)$ e aplicará um novo tom de cinza, obtendo assim a $g(x, y)$. Abaixo, exemplos típicos da função R .

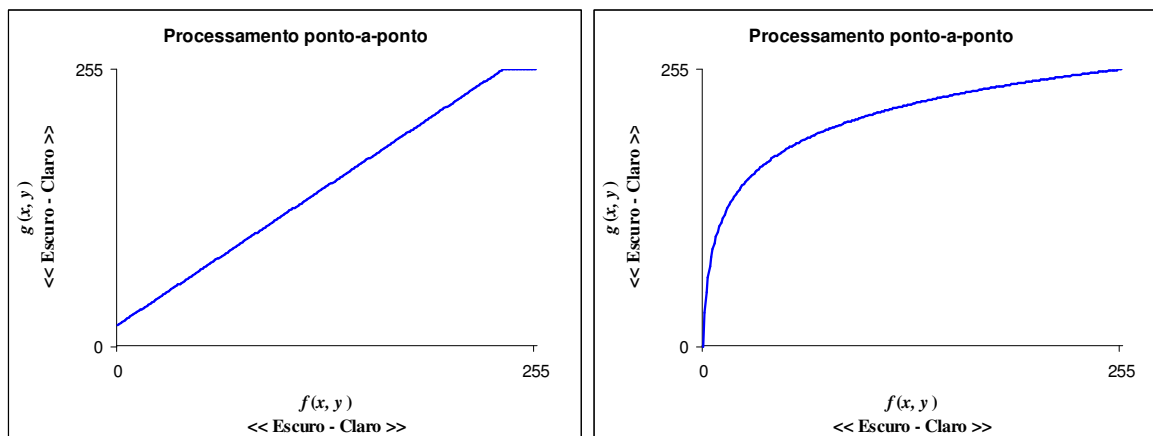


Figura 2.6 – Exemplos da função de transformação de níveis de cinza.

Processamento por máscara

Diferentemente do *processamento ponto-a-ponto*, um ponto de $g(x, y)$ é definido agora pelos valores da vizinhança de um ponto em $f(x, y)$. A máscara é uma pequena matriz bidimensional, na qual os valores de seus elementos determinam a natureza do processo. O processamento por máscara recebe o nome de *filtragem espacial*. (Gonzales; Woods, 2000). Os filtros *passa-baixas* atenuam ou eliminam componentes de alta frequência. Este tipo de filtro causa uma suavização da imagem. Os filtros *passa-altas* atenuam ou eliminam componentes de baixa frequência e realçam regiões de alto contraste. Existe também os filtros *passa-faixa*, capazes de remover ou atenuar componentes acima e abaixo da zona de corte.

2.7 – Segmentação

Etapa fundamental que define o sucesso ou fracasso da análise. A segmentação de imagens monocromáticas está baseada na mudança brusca de valores de níveis de cinza, que geralmente está associada ao contorno de objetos.

2.7.1 – Limiarização (*Thresholding*)

A limiarização é uma das mais importantes técnicas de segmentação de imagens. É através dela que se consegue separar os objetos do fundo de uma imagem. Em uma imagem monocromática (imagem em níveis de cinza), por exemplo, com dois conjuntos representando a

conectividade entre os pixels, facilmente pode-se dizer que um grupo representa o objeto na imagem e o outro grupo, representa o fundo. Portanto, os dois grupos foram separados por um *limiar* T , Figura 2.7 (a), fazendo com que o conjunto de conectividade assuma um novo valor, tornando a imagem monocromática em uma imagem binária, Figura 2.7 (c). O *limiar* T pode ser facilmente entendido pela Equação (14).

$$g(x, y) = \begin{cases} 1 & \text{se } f(x, y) > T \\ 0 & \text{se } f(x, y) \leq T \end{cases} \quad (14)$$

onde: $f(x, y)$ é a função que representa a intensidade luminosa na imagem, e $g(x, y)$, a nova função de luminosidade depois da separação do *limiar* T .

Caso a imagem tenha três ou mais grupos, pode-se usar a *limiarização multiníveis*. Nesse caso, seriam classificados como objeto, todos os valores de $f(x, y)$ entre T_1 e T_2 e o restante faria parte do fundo da imagem. A *limiarização multiníveis*, Figura 2.7 (b), é geralmente menos confiável que a de limiar único devido à dificuldade de estabelecimento de múltiplos limiares que isolem efetivamente as regiões de interesse. Problemas desta natureza são melhores resolvidos com um limiar único variável (Gonzalez e Woods, 2000).

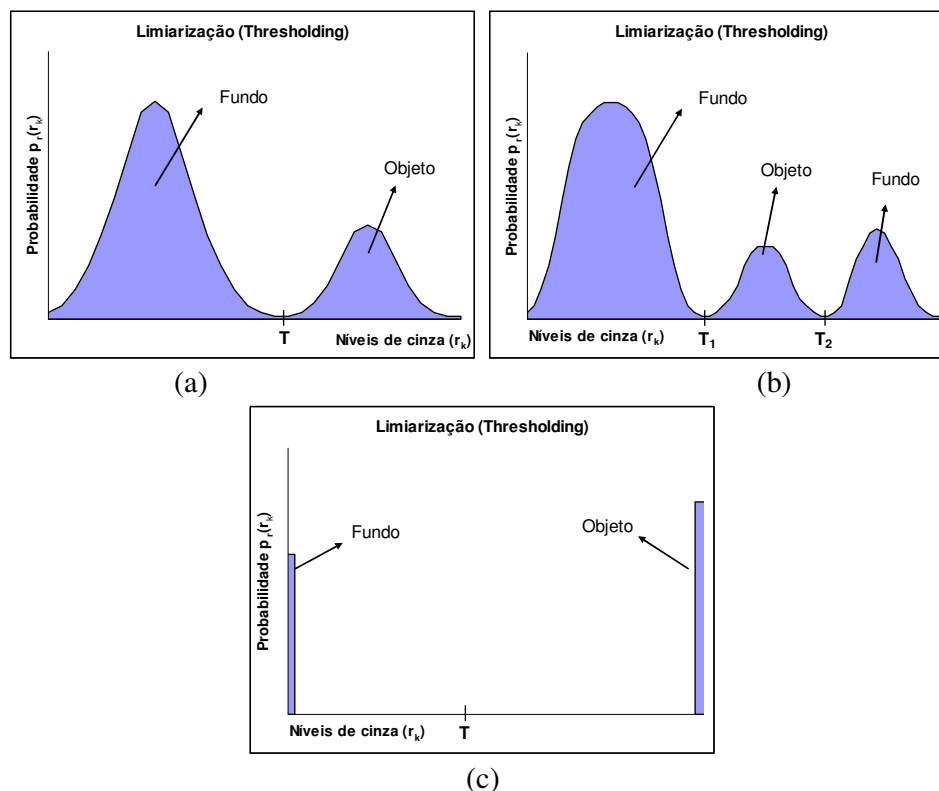


Figura 2.7 – Histograma com limiar (a) único, (b) múltiplos, (c) resultado da limiarização.

Quando o *limiar* T depender apenas de $f(x, y)$, será chamado de *limiar global*. Mas em certos casos, quando se trata de regiões que carregam o nível médio de sua vizinhança, função $p(x, y)$, T também depende de $p(x, y)$. Nessa situação o *limiar* T é chamado de *limiar dinâmico*.

2.7.1.1 – Limiarização Global Simples

Como o próprio nome sugere, esta é a técnica mais simples de limiarização, dividindo o histograma por um único limiar em duas regiões. Cada pixel é então rotulado como sendo do fundo (valor 0) ou do objeto (valor 1), dependendo do seu nível de cinza ser maior ou menor que o *limiar* T pré-estabelecido. A imagem, que anteriormente era em escala de cinza, agora passa a ser binária, preto (valor 0) ou branco (valor 1), que são os únicos valores possíveis. Apesar de esta técnica ser simples, o limiar é escolhido visualmente com base no seu histograma. Portanto, espera-se sucesso apenas em ambientes altamente controlados. Uma simples modificação na iluminação normalmente impossibilita a escolha visual correta do limiar.

2.7.1.2 – Limiarização Ótima

O objetivo dessa técnica é de eliminar a escolha arbitrária e subjetiva do limiar. Sabe-se que o histograma contém a distribuição de probabilidade de ocorrência dos níveis de cinza em uma imagem, então, pode-se calcular o limiar com base nas propriedades estatísticas. Supondo-se uma imagem com apenas duas regiões principais de brilho, o histograma fornece a probabilidade de brilho $p(z)$ global, que é a soma ou mistura de duas densidades unimodais, uma para a região clara e outra para a região escura da imagem. Neste caso, a função densidade de probabilidade da mistura é dada pela Equação (15).

$$p(z) = P_1 p_1(z) + P_2 p_2(z) \quad (15)$$

onde P_1 e P_2 são as probabilidades dos níveis de cinza e $p(z)$ é a probabilidade de brilho global.

Se a forma das densidades for conhecida ou assumida, pode-se calcular um limiar ótimo, minimizando o erro. No caso gaussiano, $p(z)$ é dado pela Equação (16).

$$p(z) = \frac{P_1}{\sqrt{2\pi\sigma_1}} \exp\left[-\frac{(z-\mu_1)^2}{2\sigma_1^2}\right] + \frac{P_2}{\sqrt{2\pi\sigma_2}} \exp\left[-\frac{(z-\mu_2)^2}{2\sigma_2^2}\right] \quad (16)$$

onde μ_1 e μ_2 são os valores médios dos dois níveis de cinza, σ_1 e σ_2 são os desvios padrão em torno das médias, P_1 e P_2 são as probabilidades dos níveis de cinza e a restrição $P_1 + P_2 = 1$ deve ser satisfeita.

Considerando que as regiões escuras correspondam ao fundo e as regiões claras correspondam aos objetos. Então, $\mu_1 < \mu_2$ e um *limiar* T pode ser definido, de modo com que

todos os *pixels* com valores iguais e abaixo de T são rotulados com valor zero e os demais com valor um. Desta maneira temos o conjunto de *pixels* que representa o fundo, com valor zero, e os *pixels* que representam o objeto, com valor um. A probabilidade $E_1(T)$ de considerar um ponto do objeto como sendo ponto de fundo é dado por

$$E_1(T) = \int_{-\infty}^T p_2(z) dz \quad (17)$$

onde $E_1(T)$ é a probabilidade de erro do objeto, T o limiar e $p_2(z)$ a probabilidade de brilho do fundo. De maneira análoga, a probabilidade $E_2(T)$ de considerar um ponto do fundo como sendo ponto do objeto é dado por

$$E_2(T) = \int_T^{\infty} p_1(z) dz \quad (18)$$

onde $E_2(T)$ é a probabilidade de erro do fundo, T o limiar e $p_1(z)$ a probabilidade de brilho do objeto. Portanto, a probabilidade total de erro é dada pela Equação (19).

$$E(T) = P_1 E_1(T) + P_2 E_2(T) \quad (19)$$

Para que o erro seja mínimo, deve-se diferenciar $E(T)$ em relação a T e igualar a zero (usando regra de *Liebnitz*). Após isso, tem-se

$$P_1 p_1(T) = P_2 p_2(T) \quad (20)$$

Aplicando esse resultado na densidade gaussiana, Equação (16), considerando manipulações e simplificações matemáticas, obtém-se a Equação (21).

$$AT^2 + BT + C = 0 \quad (21)$$

onde:

$$A = \sigma_1^2 - \sigma_2^2 \quad (22)$$

$$B = 2(\mu_1 \sigma_1^2 - \mu_2 \sigma_2^2) \quad (23)$$

$$C = \mu_1 \sigma_1^2 - \mu_2 \sigma_2^2 + 2\sigma_1^2 \sigma_2^2 \ln\left(\frac{\sigma_2 P_1}{\sigma_1 P_2}\right) \quad (24)$$

A possibilidade de duas soluções indica que dois valores de limiar podem ser utilizados para obter-se uma solução ótima. Se as variâncias forem iguais, $\sigma^2 = \sigma_1^2 = \sigma_2^2$, então apenas um limiar é necessário.

$$T = \frac{\mu_1 + \mu_2}{2} + \frac{\sigma^2}{\mu_1 - \mu_2} \ln\left(\frac{P_2}{P_1}\right) \quad (25)$$

E se as probabilidades P_1 e P_2 forem iguais.

$$T = \frac{\mu_1 + \mu_2}{2} \quad (26)$$

O mesmo resultado vale se $\sigma = 0$

A utilização deste método para a determinação do limiar ótimo, também pode ser aplicado analogamente a outras densidades unimodais de forma conhecida, como as densidades de *Raleigh* e a *log normal*.

2.7.1.3 – Método de Otsu

Este método, descrito em (Gonzalez, Woods e Eddins, 2004), foi proposto por Otsu em 1979. Consiste em achar um limiar que maximize a variância (σ^2) entre os dois grupos de pixels (fundo e objeto). Para aplicar o método de Otsu, é necessário normalizar o histograma aplicando a Equação (11). Para um determinado limiar T , têm-se dois grupos de pixels, o grupo C_0 , formado pelos valores $\{0, 1, 2, \dots, T-1\}$ e o grupo C_1 , formado pelos valores $\{T, T+1, \dots, L-1\}$. Usando a Equação (27) encontra-se a variância σ_B^2 .

$$\sigma_B^2 = w_0(\mu_0 - \mu_T)^2 + w_1(\mu_1 - \mu_T)^2 \quad (27)$$

onde:

$$w_0 = \sum_{k=0}^{T-1} p_r(r_k) \quad (28)$$

$$w_1 = \sum_{k=T}^{L-1} p_r(r_k) \quad (29)$$

$$\mu_0 = \sum_{k=0}^{T-1} k \frac{p_r(r_k)}{w_0} \quad (30)$$

$$\mu_1 = \sum_{k=T}^{L-1} k \frac{p_r(r_k)}{w_1} \quad (31)$$

$$\mu_T = \sum_{k=0}^{L-1} k p_r(r_k) \quad (32)$$

onde, σ_B^2 é a variância, w_0 é a probabilidade do grupo C_0 , w_1 é a probabilidade do grupo C_1 , μ_0 é a média do grupo C_0 , μ_1 é a média do grupo C_1 , k é o nível de cinza e μ_T é a média total do histograma. Aplicando-se a Equação (27) para todos possíveis limiares, aquele que resultar no maior valor de σ_B^2 será o escolhido.

2.8 – Influência da Iluminação

A iluminação tem um papel muito significativo no processo de limiarização, uma vez que provoca alterações significativas no histograma original. Usando o conceito de que uma imagem pode ser descrita como uma função $f(x, y)$ do produto da reflectância $r(x, y)$ pela iluminação $i(x, y)$ (Marques Filho, Viera Neto; 1999).

$$z(x, y) = \ln(f(x, y)) \quad (33)$$

$$\ln(f(x, y)) = \ln(i(x, y)) + \ln(r(x, y)) \quad (34)$$

$$z(x, y) = i'(x, y) + r'(x, y) \quad (35)$$

Considerando $i'(x, y)$ e $r'(x, y)$ como variáveis aleatórias independentes, o histograma de $z(x, y)$ é dado pela convolução do histograma $i'(x, y)$ com o de $r'(x, y)$, denotado por $i'(x, y) * r'(x, y)$.

$$i'(x, y) * r'(x, y) = \int \int_{-\infty}^{\infty} i'(\alpha, \beta) r'(x - \alpha, y - \beta) d\alpha d\beta \quad (36)$$

onde α e β são as variáveis de integração.

A solução desta equação não é relativamente simples, por isso, quando for possível o acesso a fonte de iluminação, utiliza-se uma técnica mais simples para resolver a influência da iluminação. Esta técnica consiste em projetar um padrão de iluminação sobre uma superfície refletora branca, obtendo-se

$$g(x, y) = \varphi i(x, y) \quad (37)$$

onde φ é uma constante que depende da superfície e $i(x, y)$ é o padrão de iluminação. Assim, para qualquer imagem $f(x, y) = i(x, y) r(x, y)$, obtém-se uma função normalizada $h(x, y)$,

$$h(x, y) = \frac{f(x, y)}{g(x, y)} \quad (38)$$

e conseqüentemente,

$$h(x, y) = \frac{r(x, y)}{\varphi} \quad (39)$$

Logo, se $r(x, y)$ pode ser limiarizada utilizando o limiar T , então $h(x, y)$ poderá ser segmentada usando um limiar T/φ .

2.8.1 – Técnicas de Iluminação

As técnicas de iluminação minimizam os efeitos de pouco contraste, reflexões e sombras (Grassi, 2005).

Citam-se os cinco dos principais sistemas de iluminação:

- Direta, Figura 2.8 (a). A luz é disposta diretamente acima do objeto de interesse;
- Difusa, Figura 2.8 (b). É utilizado onde as características da superfície do objeto são importantes;
- Traseira, Figura 2.8 (c). Ideal para aplicações onde o contorno dos objetos é suficiente. Produz imagem preto-e-branco;
- Estruturada, Figura 2.8 (d). Estabelece um padrão de luz conhecido na superfície de trabalho e distúrbios indicam a presença do objeto. Pode-se fazer uma análise sobre as características tridimensionais do objeto;
- Direcional, Figura 2.8 (e). Utilizada principalmente para inspeção superficial de objetos.

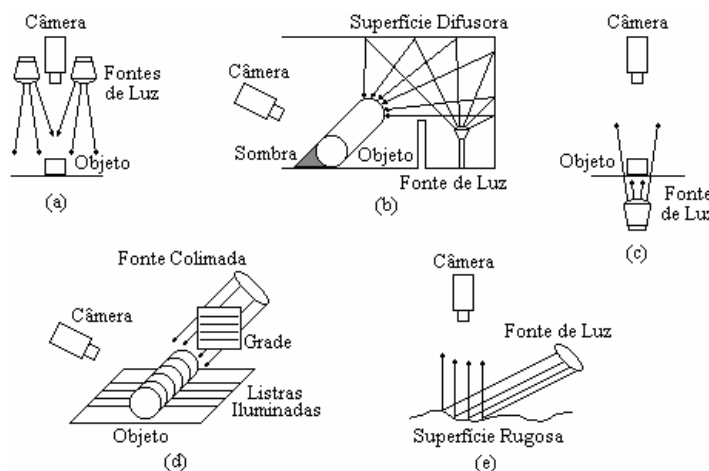


Figura 2.8 – Técnicas de Iluminação. (a) Direta, (b) Difusa, (c) Traseira, (d) Estruturada e (e) Direcional.

2.9 – Morfologia Matemática

A morfologia matemática é uma abordagem sugerida na década de 1960, na França, com a preocupação de explorar as formas presentes em imagens. Desenvolvida inicialmente para tratar imagens binárias, as primeiras aplicações foram no estudo de porosidade de minerais, sendo estendida, posteriormente, para imagens de níveis de cinza. Atualmente, há estudos no contexto de imagens coloridas (Hirata, 2002). O princípio básico da morfologia matemática consiste em extrair as informações relativas à geometria e topologia de um conjunto desconhecido (uma imagem) pela transformação através de outro conjunto completamente definido, chamado de *elemento estruturante*. Portanto, a base da morfologia matemática está estruturada na Teoria de Conjuntos. Seja, por exemplo, o conjunto de todos os *pixels* pretos de uma imagem binária. Este conjunto pertence ao espaço bidimensional de números inteiros \mathbf{Z}^2 e é uma descrição completa desta imagem. Cada elemento é um vetor bidimensional cujas coordenadas são as coordenadas (x, y) dos *pixels* pretos na imagem. Imagens digitais em níveis de cinza podem ser representadas por conjuntos cujas componentes estão em \mathbf{Z}^3 . Nesse caso, duas componentes de cada elemento do conjunto se referem às coordenadas (x, y) do *pixel*, enquanto o terceiro corresponde ao valor discreto de intensidade. Conjuntos em espaços de maiores dimensões podem conter outros atributos de imagens, como cor e componentes que variem com o tempo. O mapeamento da morfologia binária pode ser expresso através de operações elementares sobre conjuntos e mais dois operadores, conhecidos como *erosão* e *dilatação*.

2.9.1 – Erosão

Sejam A e B conjuntos no espaço \mathbf{Z}^2 . A erosão de A por B , denotada $A \ominus B$, é definida como:

$$A \ominus B = \{x | (B)_x \subseteq A\} \quad (40)$$

Ou seja, a erosão de A por B resulta no conjunto de pontos x tais que B , transladado de x , está em A . Desta maneira, o conjunto A representa o conjunto de *pixels* da imagem e o conjunto B , representa o *elemento estruturante*.

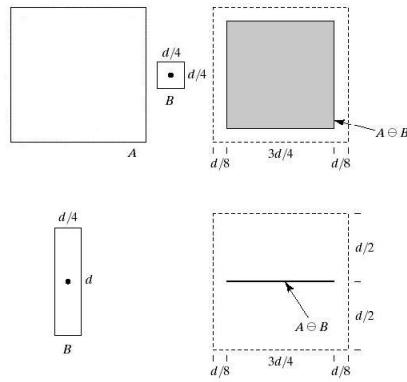


Figura 2.9 – Erosão de A por B.

2.9.2 – Dilatação

Sejam A e B conjuntos no espaço Z^2 e \emptyset o conjunto vazio, define-se a dilatação de A por B, denotada por $A \oplus B$, como:

$$A \oplus B = \left\{ x \left| \left(\hat{B} \right)_x \cap A \neq \emptyset \right. \right\} \tag{41}$$

O processo de dilatação começa na obtenção da reflexão de B em torno de sua origem, seguido da translação dessa reflexão por x. A dilatação de A por B é então o conjunto de todos os deslocamentos x tais que \hat{B} e A sobreponham-se em pelo menos um elemento não nulo. Essa interpretação permite que a Equação (41) seja reescrita como:

$$A \oplus B = \left\{ x \left| \left[\left(\hat{B} \right)_x \cap A \right] \subseteq A \right. \right\} \tag{42}$$

Novamente, o conjunto A representa o conjunto de pixels da imagem e o conjunto B, o *elemento estruturante*.

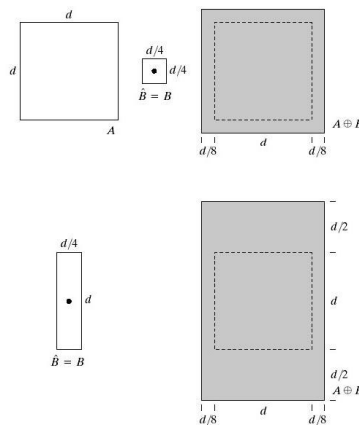


Figura 2.10 – Dilatação de A por B.

2.9.3 – Abertura e Fechamento

Geralmente, o processo de *abertura*, suaviza o contorno de uma imagem, quebra istmos e elimina proeminências delgadas. O *fechamento*, por sua vez, funde as quebras em golfos finos, elimina pequenos buracos, preenche fendas em um contorno e também tende a suavizar os contornos. A Equação (43) define o processo de *abertura* de A por B , que é simplesmente a erosão de A por B seguido da dilatação do resultado por B .

$$A \circ B = (A \ominus B) \oplus B \quad (43)$$

O *fechamento* de A por B , é simplesmente a dilatação de A por B seguido da erosão do resultado por B , com mostra a Equação (44).

$$A \bullet B = (A \oplus B) \ominus B \quad (44)$$

2.9.4 – Extração de Fronteiras

Seja $\beta(A)$ a fronteira de um conjunto A . Podemos demonstrar que $\beta(A)$ pode ser obtido através da erosão de A por um *elemento estruturante* B específico, seguido da diferença de conjuntos entre A e a sua erosão. Desta maneira, podemos escrever a Equação (45).

$$\beta(A) = A - (A \ominus B) \quad (45)$$

onde $\beta(A)$ = fronteira do conjunto A ; A = conjunto de pixels e B = elemento estruturante

Considerando-se a Figura 2.11, composta pelo conjunto A e pelo *Elemento Estruturante* B .

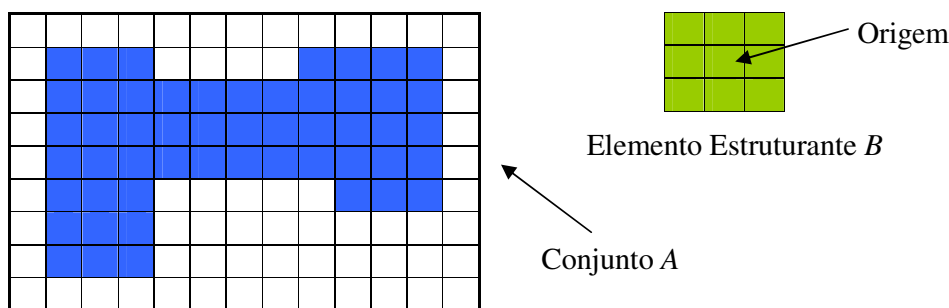


Figura 2.11 – O conjunto de *pixels* A e o Elemento Estruturante B .

O conjunto A representa uma imagem, neste caso com tamanho exagerado dos pixels, bastante simplificada. Vamos fazer com que o centro do *Elemento Estruturante* B percorra todos os pixels da imagem, começando pelo canto superior esquerdo, Figura 2.12, por exemplo.

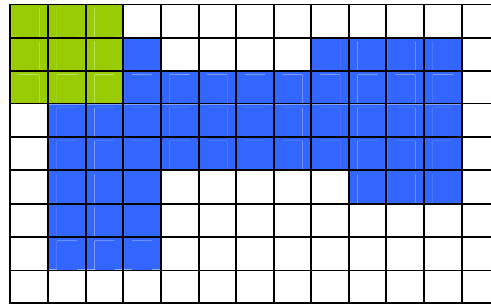


Figura 2.12 - Início do processo de erosão.

Quando o *Elemento Estruturante B* estiver contido no conjunto *A*, o pixel do conjunto *A* coincidente com o centro do *Elemento Estruturante B*, é rotulado como pertencente ao conjunto resultado da erosão de *A* por *B*, Figura 2.13.

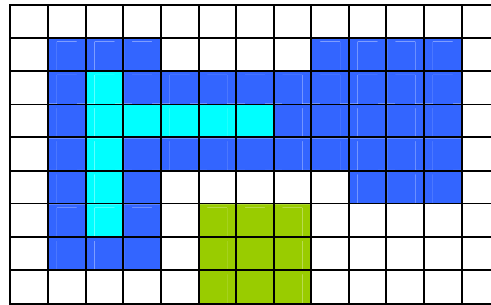


Figura 2.13 – O processo erosão e *pixels* rotulados.

Calculando a diferença entre o conjunto *A* e o resultado da erosão de *A* por *B*, Figura 2.14 (a), obtendo-se o contorno do conjunto *A*, Figura 2.14 (b).

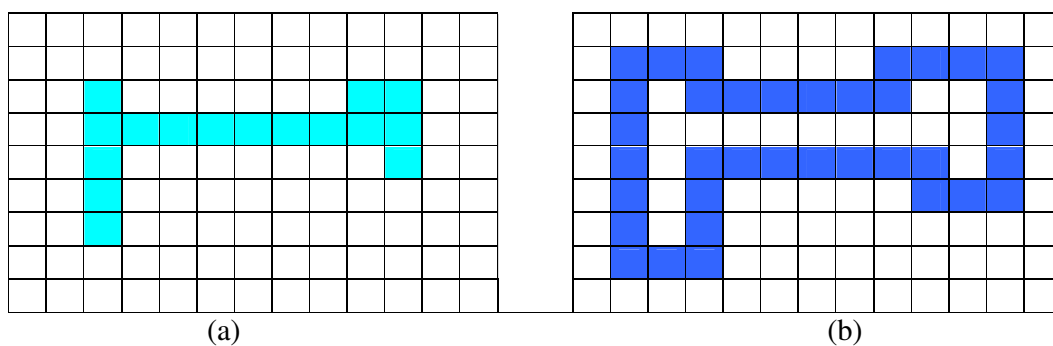


Figura 2.14 - Resultado do processo e erosão (a) e o contorno obtido (b).

Utilizando outro *Elemento Estruturante* e repetindo o processo no conjunto *A*, observa-se que este tipo de *Elemento Estruturante* gera contornos mais simplificados.

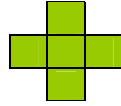


Figura 2.15 - *Elemento Estruturante.*

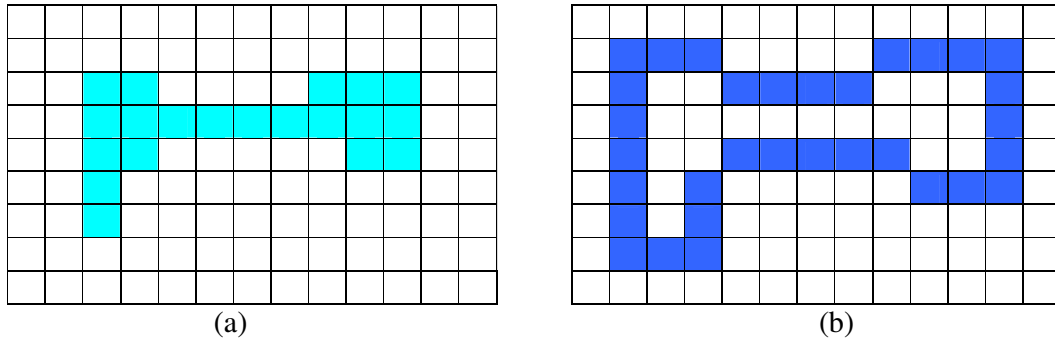


Figura 2.16 - Resultado do processo de erosão (a) e o contorno obtido (b).

3 – DESENVOLVIMENTO DO SISTEMA

3.1 – Equipamentos e Programas Utilizados

Os equipamentos foram selecionados em função da disponibilidade de recursos e infraestrutura instalada no Laboratório de Robótica da Universidade Federal do Rio Grande do Sul.

3.1.1 – Câmeras

A câmera CCD utilizada foi a *SAMSUNG BW-2302ED*. Suas características estão listadas na Tabela 2 do Anexo I, utilizada com resolução de 720x480 *pixels*.



Figura 3.1 – Câmera CCD.

Uma *webcam* também foi utilizada, com a vantagem inicial de dispensar placa de captura de imagem devido a sua conexão USB (*Universal Serial Bus*). O modelo utilizado foi a *Genius VideoCAM Express V2*, com resolução de 352x288 *pixels*.

3.1.2 – Computador

Foi utilizado um *Pentium IV 3.0 GHz HT* com 1024 MB de memória RAM, placa de aquisição de imagens VTV2004 (Pico2000) e Sistema Operacional *Windows XP Professional*.

3.1.3 – Placa de Aquisição de Imagem

A placa para captura de imagens corresponde a uma VTV2004. O mesmo modelo foi utilizado por (Grassi, 2005) e (Manozzo, 2005). Esta placa é normalmente utilizada em CFTV (Circuito Fechado de Televisão) baseado em PC, sendo compatível com *slot PCI*, captura com 30 fps e possui entradas para até quatro câmeras com conexão através de um cabo coaxial BNC. Uma mudança do *driver* original foi necessária para tornar a placa disponível para a aplicação implementada. No site <http://btwincap.sourceforge.net/download.html> está disponível o arquivo *btwdmrvinstaller5.3.8.zip* que contém o executável *Btinstaller.exe*. Para a instalação, deve-se escolher a opção n° 24 (*Generic*).

3.1.4 – Software MATLAB

O *MATLAB*, *MATrix LABoratory* (Laboratório de Matrizes), é um software programável da “*The MathWorks*”, que segundo seu criador, *Cleve Moler*, é um ambiente de computação técnico-científica para desenvolvimento de sistemas sofisticados e eficientes. Sua linguagem de

programação é de alto nível baseada em matrizes. “Desenvolver soluções no *MATLAB* é, sem dúvida, muito mais eficiente e rápido do que em qualquer outra linguagem de programação convencional como *C/C++*, *Fortran* ou *Visual Basic*.” (Matsumoto, 2004). No *MATLAB*, todas as variáveis são dimensionadas automaticamente ao serem referenciadas pela primeira vez e seus valores permanecem na memória de trabalho até que sejam eliminadas. Além disso, o *MATLAB* apresenta bibliotecas de rotinas específicas (*toolbox*) para cada aplicação, inclusive para o processamento de imagens. Todas as funções pré-definidas em uma biblioteca podem ser modificadas, isto significa que pode-se escrever funções específicas para o problema em questão. Estas funções, novas ou não, são armazenadas em arquivos denominados “*m-files*” (arquivos-*m*) e podem ser utilizados em aplicações diversas. Uma característica importante e útil do *MATLAB*, é que os programas escritos podem ser transformados em programas executáveis independentes do ambiente *MATLAB*. O “*MATLAB Compiler*” gera o código-fonte *C/C++* a partir de funções contidas nos *m-files* e incorpora automaticamente as bibliotecas necessárias. Porém, a utilização de um compilador *C/C++* (por exemplo, *MS-Visual C++* ou *Borland C++*) é necessário para a geração do arquivo executável. Tendo conhecimento destas características do *MATLAB* e de que o modelo proposto de uma imagem digital é uma matriz de $M \times N$ pixels, além do conhecimento de outros trabalhos técnico-científicos na área de processamento de imagens, fica então justificado a utilização do *MATLAB* para resolução do problema proposto neste trabalho.

O *MATLAB* possui uma convenção diferente de eixos para imagem digital. Os eixos (x, y) são denotados por (r, c), linha (*row*) e coluna (*column*). A origem é representada por (r, c) = (1, 1). (Gonzales; Woods e Eddins, 2004) – Figura 3.2. Essa alteração também afeta a matriz digital.

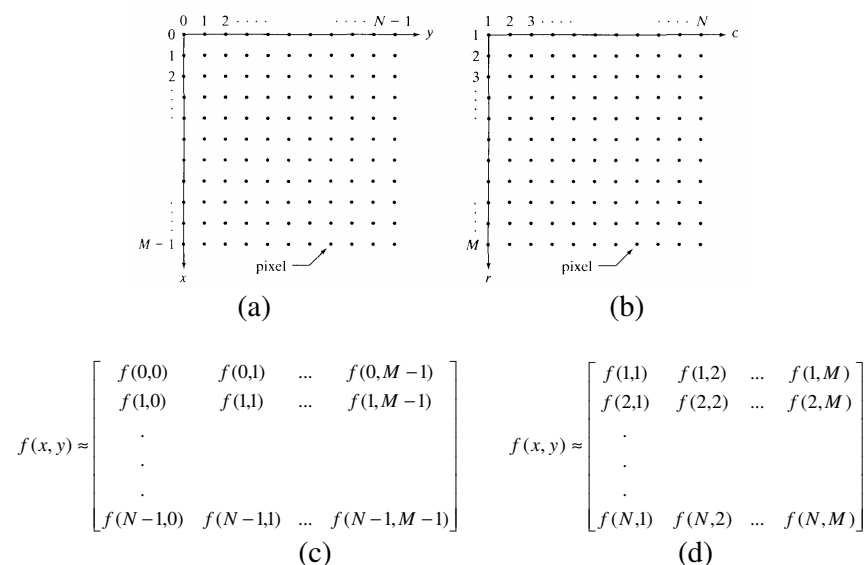


Figura 3.2 – Convenções utilizadas pela literatura (a) e (c), pelo *MATLAB* (b) e (d).

O *Simulink* é uma interface gráfica do *MATLAB*, onde pode-se simular os efeitos dinâmicos dos sistemas. As funções que antes eram referenciadas através de *m-files*, passam agora a ser referenciadas através de blocos, com os mesmos parâmetros editáveis.

3.1.5 – Dispositivo de Comunicação - *Hardware IRBCom*

O *IRBCom* foi desenvolvido por Bayer (Bayer, 2004), como sendo uma alternativa a baixo custo, para fazer a comunicação bidirecional entre o robô *ABB IRB1400* e um microcomputador utilizando a porta paralela do microcomputador e uma placa E/S digital *ABB DSQC 223*, com 16 canais digitais de entrada e 16 de saída para sinais discretos, originalmente destinada ao controle de dispositivos integrados ao robô. 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 segurança de operação, garantindo isolamento galvânico entre as duas interfaces para tensões da ordem de até 1,5kV. (Bayer, 2004).

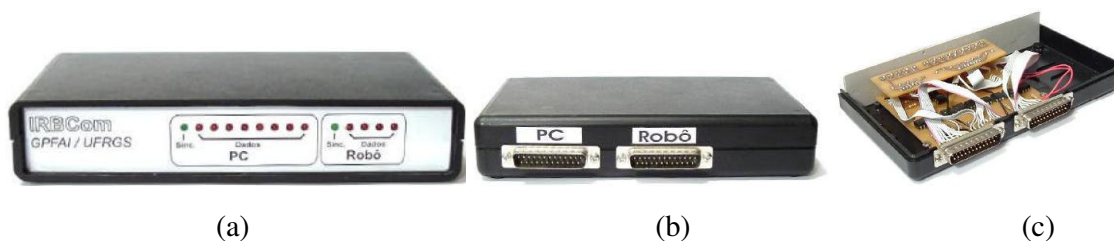


Figura 3.3 – *IRBCom*, painel frontal (a), conectores externos (b) componentes internos (c).

Para garantir a flexibilidade na utilização do *IRBCom*, Bayer desenvolveu rotinas compiladas em formato *dll* (*Dinamic Library Link*), as quais permitem a integração do Sistema *IRBCom* com programas escritos nas principais linguagem de programação (*C*, *Visual Basic*, *Pascal* e *Delphi*), destinados ao Sistema Operacional *Microsoft Windows* de 32 bits. O Sistema de comunicação do *IRBCom* opera no sentido PC (*cliente*) – Robô (*servidor*), sendo que a comunicação no sentido contrário só ocorre sob solicitação do *cliente*. O programa principal em execução no controlador do robô deve prever um ciclo contínuo que faz a leitura no comando, identifica e ativa a rotina solicitada e que em função das coordenadas (*x*, *y*) recebidas no robô, executa o movimento correspondente. No intervalo de tempo em que a tarefa está sendo executada, o controlador é bloqueado para receber outros comandos, ficando o PC aguardando até que o controlador esteja disponível. Este tipo de comunicação é viável no caso de tarefas de manipulação de objetos (*pick-and-place*). O *IRBCom* não apresenta a possibilidade de envio e ativação de programas, o que seria o modo mais indicado para a proposta deste trabalho. O

sistema opera com taxa de transmissão de dados no sentido PC/Robô de 12,5 bytes/s e Robô/PC em 6,1 bytes/s. A figura 3.5, mostra um diagrama simplificado do fluxo de informações.

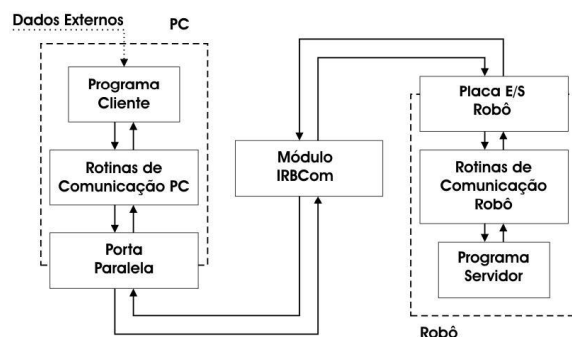


Figura 3.4 – Fluxograma de informações.

Sobre a taxa de transmissão, Bayer afirma: “O envio de trajetórias mais refinadas se mostra possível, porém com um baixo rendimento, devido à necessidade de interrupção do movimento para a leitura dos dados do próximo comando, impossibilitando assim um movimento suave e com altas velocidades.” Neste trabalho, é utilizado o *IRBCom*, pois entende-se que em uma aplicação industrial, o robô deva ser dotado de uma interface de comunicação de dados desenvolvida pelo próprio fabricante, e também porque não é o objetivo deste trabalho desenvolver um sistema de comunicação.

3.1.6 – Robô Modelo *ABB IRB1400*

O robô utilizado foi o *IRB1400* da *ABB*. Apresenta seis graus de liberdade com juntas rotacionais. A semelhança de seus movimentos com o braço humano lhe confere o nome de robô antropomorfo. Possui um amplo volume de trabalho que pode ser comparado a uma porção de esfera. Devido às suas juntas rotacionais, o robô antropomorfo pode atingir um mesmo ponto com múltiplas configurações de seus eixos, possibilitando contornar obstáculos presentes no volume de trabalho. Sua linguagem de programação é a *RAPID*, específica para robôs da *ABB*.

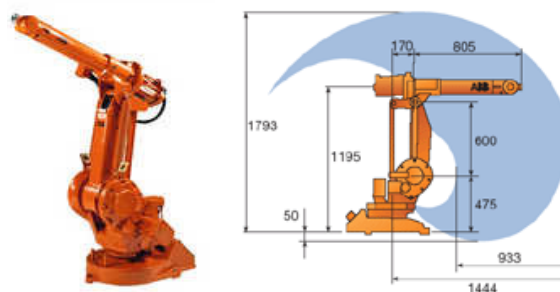


Figura 3.5 – Robô modelo - *ABB IRB1400*.

Nem todas as aplicações requerem a versatilidade de um robô antropomorfo, mas em geral, o robô antropomorfo consegue realizar todas as tarefas com maior facilidade quando comparado com robôs com juntas prismáticas, o que justifica seu custo elevado. Outras características do robô são descritas na Tabela 1 do Anexo I.

3.1.7 - Software Boundary Lines

Com o objetivo de tornar o processamento automático e simples para o usuário, foi desenvolvido um software de controle, em *Visual Basic.NET*, chamado “*Boundary Lines*”, interface na Figura 3.6. O *Boundary Lines* é responsável pelo controle do *MATLAB*, manipulação dos resultados, geração do programa *RAPID* e envio de dados para o *IRBCom*. Basicamente, o *Boundary Lines* possui três métodos de operação: *Simulação*, que pode ser feita usando o *MATLAB* ou *Simulink*, *Módulo Semi-Automático*, onde o resultado do processamento é um programa em linguagem *RAPID* e *Módulo Automático*, onde os pontos de contorno do objeto são transmitidos via *IRBCom* para o robô. Utilizando a operação de *Simulação*, do “*Controle Principal*”, consegue-se verificar imediatamente o sucesso ou fracasso da segmentação da imagem. A variação de parâmetros tais como iluminação e contraste podem gerar coordenadas incorretas. Lembrando que os sistemas artificiais de visão funcionam muito bem apenas para as situações que foram programadas, se o resultado do processo de segmentação não for o esperado, deve-se então alterar a iluminação ou o contraste entre o objeto e o fundo da imagem.

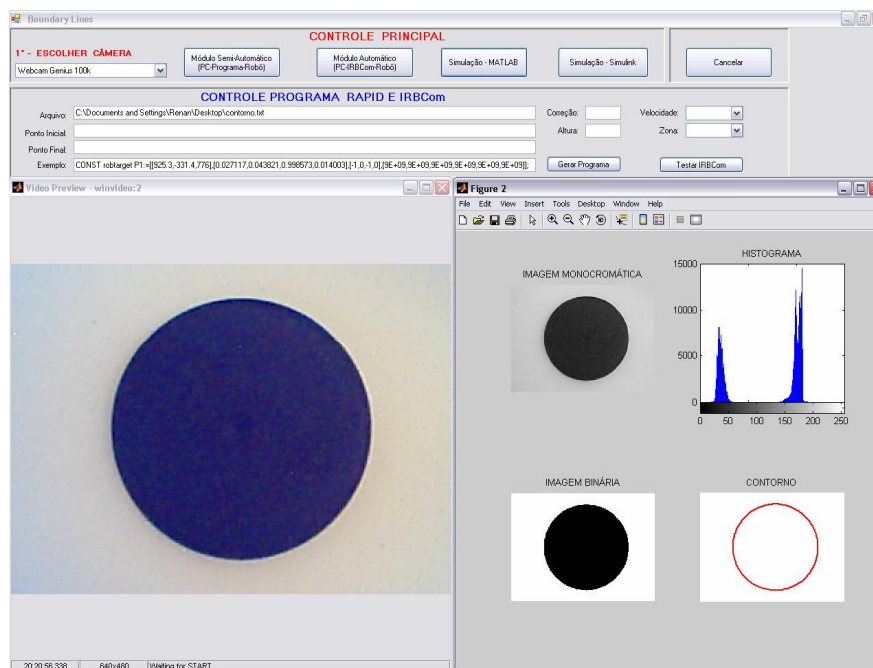


Figura 3.6 – Interface do *Boundary Lines*.

A simulação detecta possíveis falhas na obtenção do contorno quando o objeto e o fundo possuem níveis de cinza muito próximos. A *Simulação* pode ser feita no *MATLAB* ou *Simulink*, não há diferença significativa entre eles, exceto que no *MATLAB* a aquisição da imagem é feita em intervalos de tempo pré-determinados e ignora regiões isoladas com menos de 100 *pixels* (este número pode ser ajustado). No *Simulink*, a aquisição ocorre em tempo real e o modelo pode ser totalmente alterado, o que não é possível no *MATLAB*.

O *Módulo Semi-Automático* utiliza as mesmas rotinas da *Simulação*, porém, gera automaticamente um programa escrito em linguagem *RAPID*. Este programa poderia ser carregado diretamente na memória do robô se o mesmo possuísse a interface de comunicação fornecida pela *ABB*. Neste caso, não há um programa sendo executado na memória do robô, como é o caso do *Módulo Automático*, onde sua transmissão de dados via *IRBCom* só é possível se houver um programa em execução no robô que aguarde os pontos. O *Módulo Automático*, também utiliza as mesmas rotinas da *Simulação*.

3.2 – Descrição e utilização do Aplicativo *Boundary Lines*

O Fluxograma das rotinas principais é mostrado na Figura 3.7

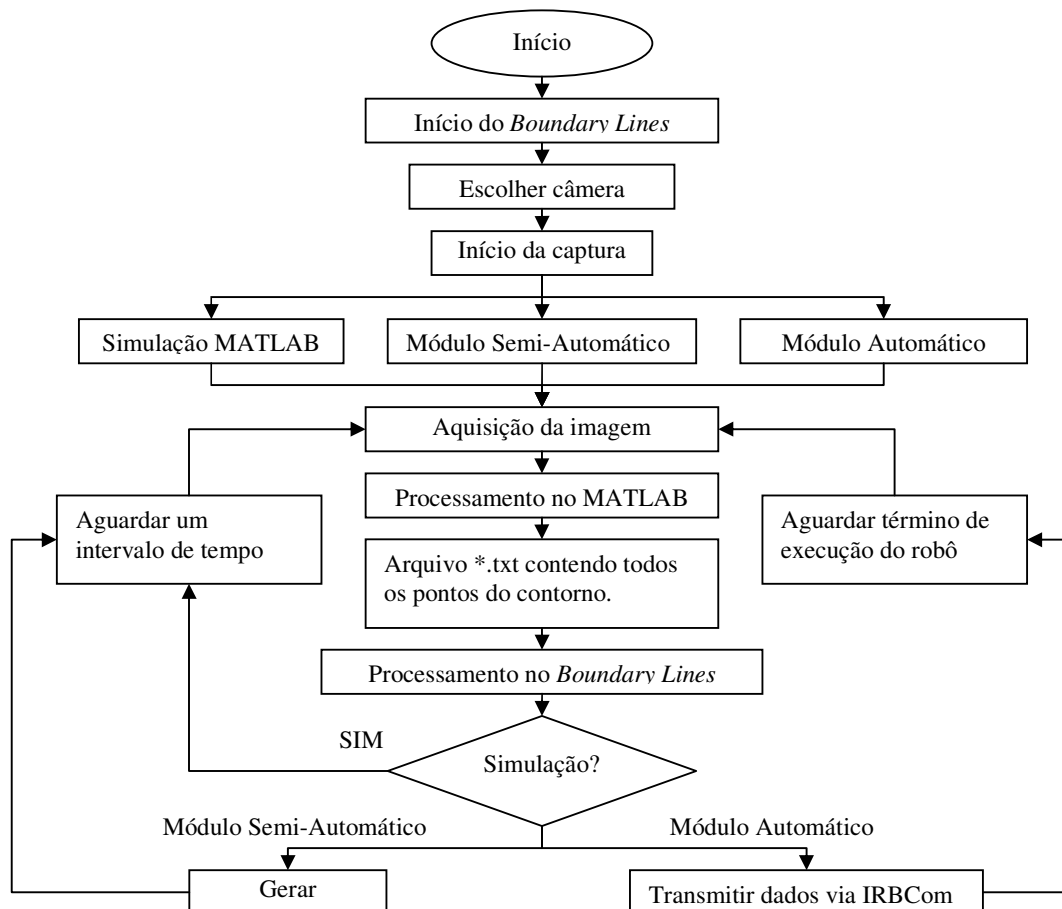


Figura 3.7 – Fluxograma das rotinas principais no *Boundary Lines*.

Primeiramente, o usuário deve escolher com qual câmera deseja realizar a aquisição. Neste caso, apenas duas câmeras estão disponíveis, a *Samsung BW2302ED* e a *Genius Webcam 100k*. Definida a câmera, seleciona-se uma das três opções de operação: *Módulo Semi-Automático*, *Módulo Automático* e *Simulação*, este último subdividido em *Simulação – MATLAB* e *Simulação – Simulink*. Através da opção “*Cancelar*” à direita do “*Controle Principal*”, na interface do aplicativo (Figura 3.6), pode-se interromper o processo em execução a qualquer momento.

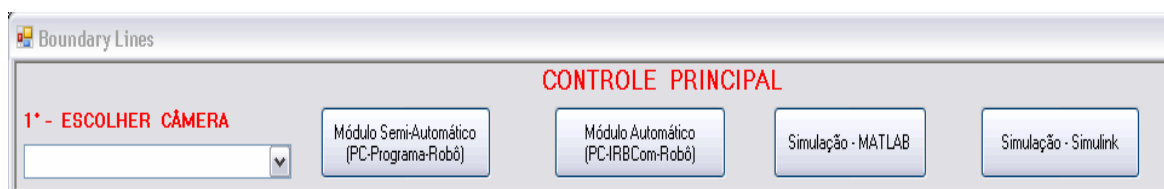


Figura 3.8 – Controle Principal do Boundary Lines.

No “*Controle Programa RAPID e IRBCom*” (Figura 3.9), é possível fazer alterações de execução do programa. Pode-se alterar parâmetros de velocidade, zona e altura, além alterar o diretório do arquivo com extensão “.txt” gerado pelo *MATLAB*, que o *Boundary Lines* utiliza para enviar os pontos ou gerar o programa. O “*Ponto inicial*” refere-se ao ponto de coordenadas (0, 0) da imagem coincidente com o ponto de referência do robô. Assim como na imagem, todos os pontos possuem uma distância em *pixels* em relação à origem, o mesmo pode-se fazer com o plano de trabalho do robô, porém com uma correção necessária. Considerando o *pixel* com proporções 1:1, ou seja, um quadrado, é necessário informar a dimensão lateral do *pixel* em milímetros no campo “*Correção*”, pois cada ponto gerado pelo *MATLAB*, é a distância (*x*, *y*) do *pixel* em relação à origem da imagem em *pixels*, que se passados diretamente para o robô, seriam entendidos em milímetros.

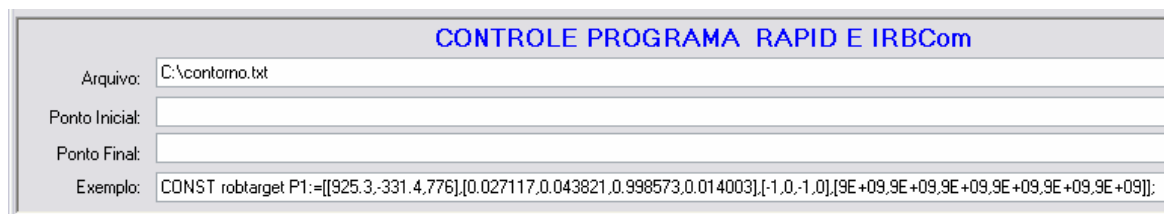


Figura 3.9 – Interface do Controle Programa *RAPID e IRBCom*.

O campo “*Altura*” significa quantos milímetros o robô se afastará perpendicularmente do plano de trabalho, opção ilustrada na figura 3.10. O *Boundary Lines* possui ainda duas opções, o

“*Testar IRBCom*”, que apenas testa a conexão entre o computador, *IRBCom* e Robô, e o “*Gerar Programa*”, utilizado após a *Simulação – MATLAB*, para gerar o programa em *RAPID*. A lógica geral do “*Simulink*” é ilustrada no diagrama de blocos da Figura 3.11.

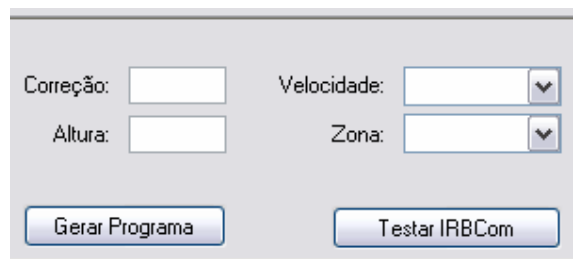


Figura 3.10 – Possibilidade de alteração de parâmetros.

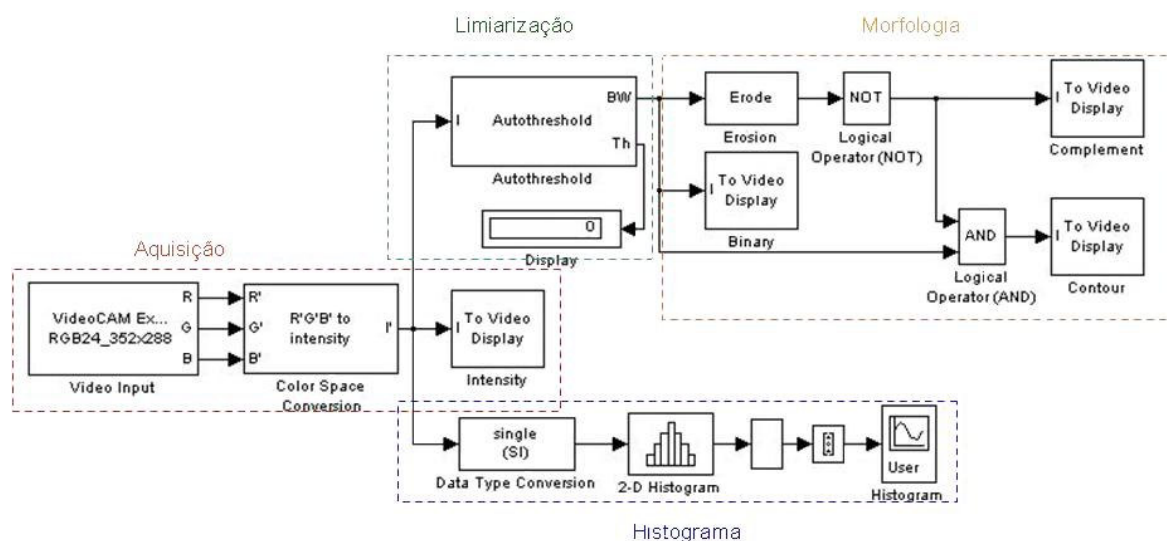


Figura 3.11 – Blocos da *Simulação – Simulink*.

Conforme a Figura 3.11, o conjunto de blocos “*Aquisição*” é responsável pela captura da imagem. Pode-se alterar, entre outros parâmetros, a resolução do vídeo e a câmera. O bloco “*Color Space Conversion*”, é responsável pela conversão de uma imagem colorida no padrão RGB em uma imagem de níveis de cinza (intensidade). O bloco “*Intensity*”, apenas visualiza o resultado da captura. O conjunto de blocos “*Histograma*” é utilizado para gerar o histograma, com uma conversão de dados de “*double*” para “*single*” por exigência do bloco “*2-D Histogram*”. Os dois blocos subsequentes organizam os dados para serem visualizados pelo bloco “*Histogram*”. O conjunto de blocos “*Limiarização*” ilustra o procedimento utilizado no *Simulink* para obter-se o contorno dos objetos. A imagem é monocromática quando chega ao bloco “*Autothreshold*”. O Método de Otsu é utilizado para fazer a limiarização, transformando a

imagem monocromática em binária. O *limiar* normalizado utilizado é mostrado no “*Display*”. No conjunto de blocos “*Morfologia*”, o bloco “*Erosion*” determina o elemento estruturante, optando-se por *4-conectados* ou *8-conectados*. O contorno é obtido através de uma operação lógica “*AND*” entre o resultado da limiarização com o complemento do resultado da erosão. As Figuras 3.12 e 3.13 exemplificam a obtenção do contorno dos objetos.

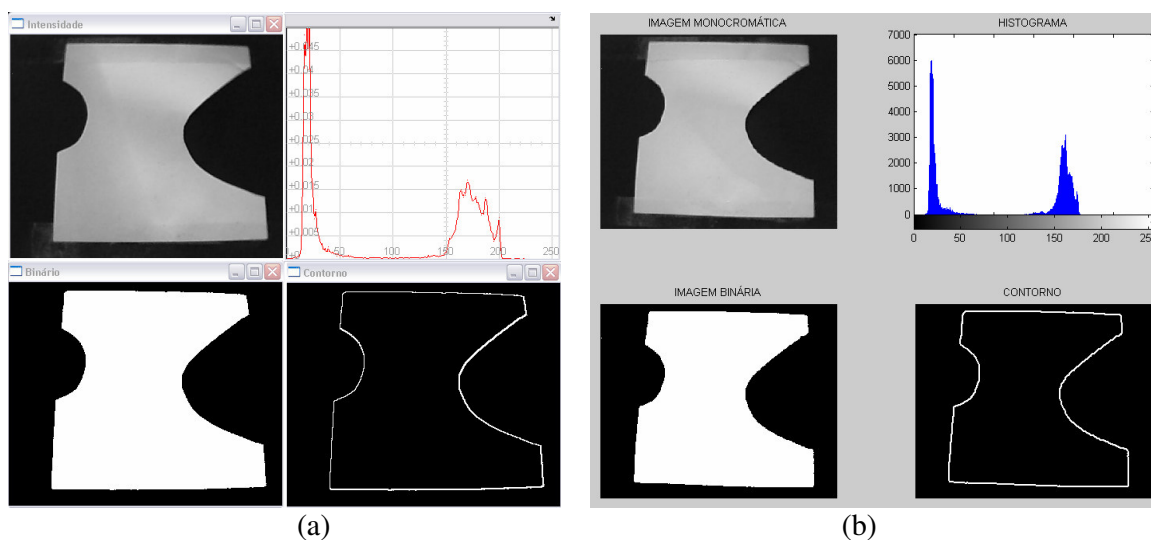


Figura 3.12 – Sucesso no processo de segmentação (a) *Simulink* e (b) *MATLAB*.

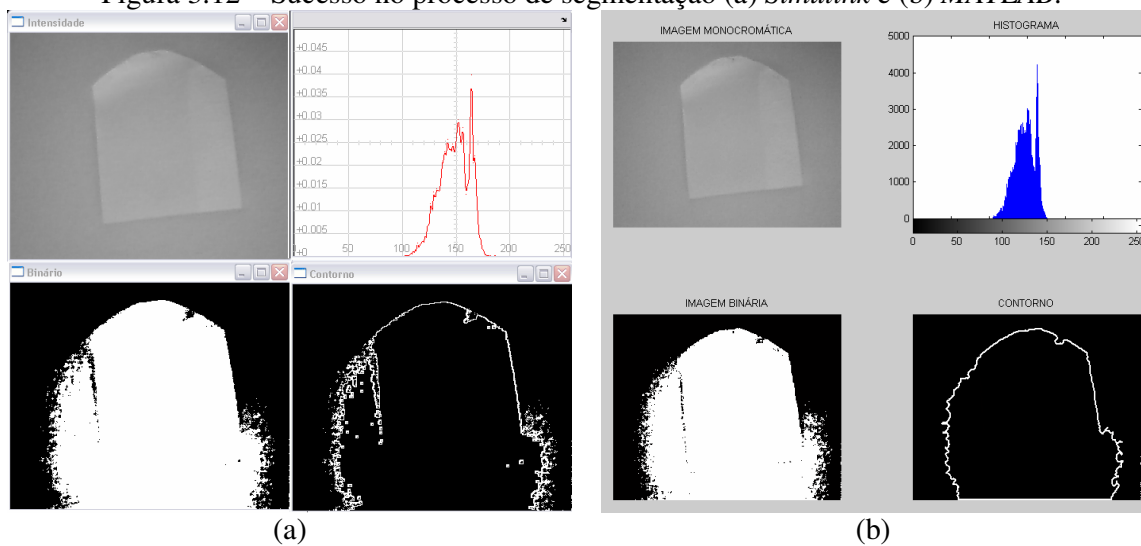


Figura 3.13 – Falha no processo de segmentação (a) *Simulink* e (b) *MATLAB*.

3.2.1 – Identificação do Fundo

A definição de um objeto claro é subjetiva e dependente do fundo, sendo assim, um objeto é considerado claro se o seu nível de cinza é maior que o do fundo, mas o mesmo objeto pode ser considerado escuro se o seu nível de cinza for menor que o do fundo. A lógica de reconhecimento de contornos de objetos do *Boundary Lines* está baseada em uma imagem com

fundo escuro e objeto claro após a limiarização. Para imagens com o fundo claro é necessário ter o complemento destas imagens, isto é, o preto passa a ser branco e o branco passa a ser preto.

Com o objetivo de criar um sistema que seja o mais independente possível da intervenção humana, define-se como “*Área de Captura*” a área total de abrangência da câmera que subdivide-se em “*Área de Teste de Fundo*” e “*Área Ótima de Reconhecimento*” – Figura 3.14.

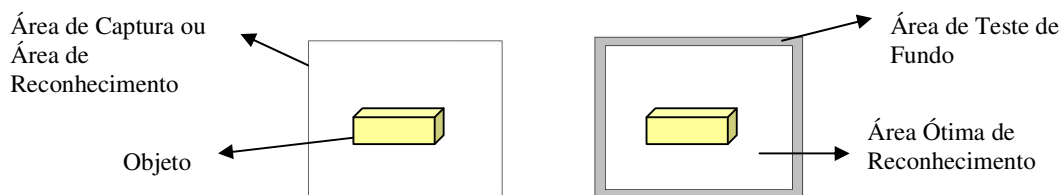


Figura 3.14 – Áreas de Captura, Reconhecimento e de Teste de Fundo.

A *Área de Teste de Fundo* é responsável pela determinação do fundo. Com uma espessura de 10 *pixels*, é feita a contagem de *pixels* pretos e brancos, sendo que o maior deles caracteriza o fundo. Caso o número seja igual, repete-se o algoritmo eliminando um *pixel* de espessura e assim sucessivamente. Após a identificação do fundo, sua área passa a fazer parte da *Área de Reconhecimento*.

A *Área Ótima de Reconhecimento* é a melhor região para reconhecer o contorno de objetos, pois não interage com o reconhecimento do fundo. A Figura 3.15 exemplifica o processo de reconhecimento. Para evitar maiores alterações nas rotinas de morfologia matemática do *Boundary Lines*, optou-se por um sistema onde os *pixels* pretos caracterizam o fundo e os *pixels* brancos, o objeto. Assim, depois de identificado o fundo, obtém-se o complemento desta imagem se o fundo for claro.

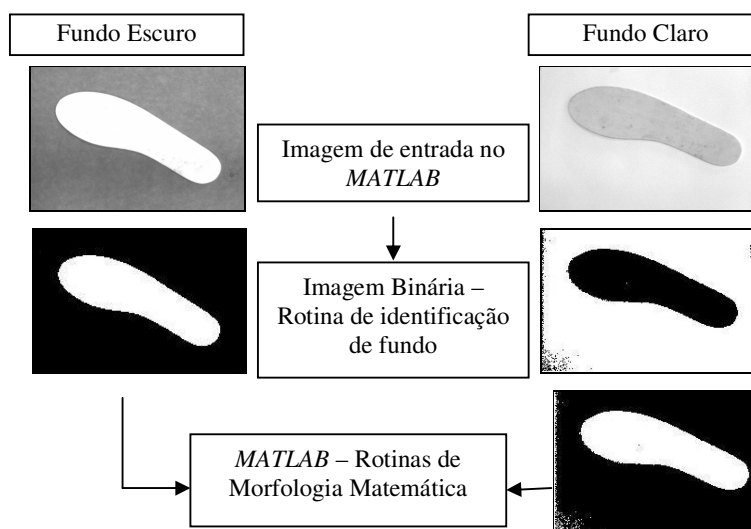


Figura 3.15 – Exemplo de identificação de fundo.

3.3 – Validação Experimental do Sistema

A validação do sistema foi realizada através de testes experimentais em laboratório.

3.3.1 – Objetivos dos Testes

Nos testes realizados com o Sistema de Visão proposto através do uso do *Boundary Lines* integrado com o *MATLAB*, todas as imagens dos objetos foram adquiridas estaticamente, ou seja, com o objeto sem deslocamento em relação à câmera. Os seguintes parâmetros foram testados.

- Capacidade de obtenção do contorno de objetos distintos em posições diferentes, incluindo a variação de contraste entre o objeto e fundo;
- Capacidade de trabalhar com câmeras diferentes e com resoluções diferentes.

3.3.2 – Bancada de Testes

A Bancada de Testes foi montada conforme ilustrado na Figura 3.16. A câmera foi posicionada de maneira perpendicular ao plano de trabalho. Desta forma, a imagem obtida é sempre uma imagem de topo dos objetos. A iluminação utilizada foi a direta para poder proporcionar variações de contraste entre o fundo e o objeto.

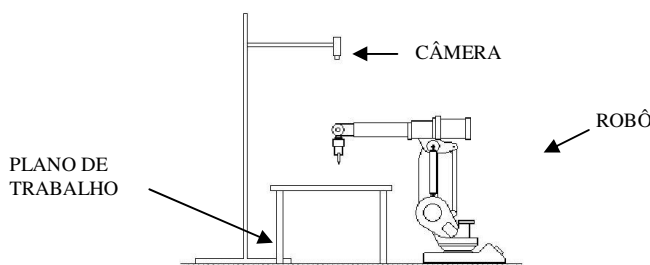


Figura 3.16 – Bancada de Testes.

A alteração dos eixos, necessária devido a posição da câmera em relação à área de trabalho do robô, é ilustrada na Figura 3.17. A orientação dos eixos do plano de trabalho do robô e da imagem é diferente, significando que qualquer alteração no eixo y da imagem corresponda a uma alteração no eixo X do plano do robô e o mesmo ocorre com o eixo x da imagem em relação ao Y do plano do robô.

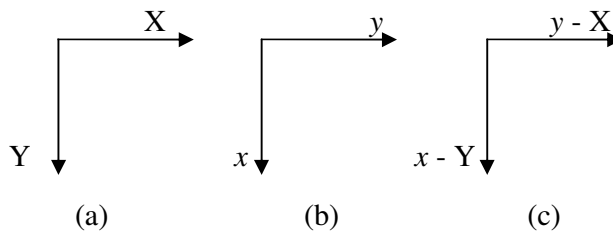


Figura 3.17 - Orientação dos eixos do plano de trabalho do robô (a), do plano da imagem (b) e eixos interpretados pelo *Boundary Lines* (c).

O plano de trabalho do robô não precisa ser coincidente com o plano da imagem. Pode-se trabalhar com planos transladados e ou rotacionados, cuidando sempre a relação de transformação entre os eixos. O *Boundary Lines* está preparado para trabalhar com planos coincidentes ou não, cuja transformação final dos eixos corresponda ao da Figura 3.20 (c).

3.3.3 – Objetos Testados

A Figura 3.21 mostra os objetos e os fundos (preto, pardo e branco) testados. Cada objeto foi testado separadamente, variando-se o fundo.



Figura 3.18 – Objetos e fundos testados.

3.3.4 – Calibração do Sistema

As exigências de calibração da câmera são em relação ao seu posicionamento. A câmera deve ser posicionada perpendicularmente ao objeto, de maneira que a imagem obtida seja a projeção da vista superior deste objeto.

A calibração do plano de trabalho do robô e do plano da imagem pode ser feita utilizando relações óticas. Relaciona-se o tamanho do objeto com a quantidade de *pixels* na mesma direção, obtendo-se o tamanho do *pixels* em milímetros. Um método alternativo pode ser usado quando não se conhece a distância focal da câmera. Consiste em obter uma imagem de um objeto com dimensões conhecidas, determinar quantos *pixels* foram necessários para representar esse objeto em uma direção. Após, utilizando uma simples relação de comprimento, determina-se o tamanho em milímetros de um dos lados do *pixel*. O tamanho do *pixel* deve ser informado no campo “correção” do *Boundary Lines*.

Na utilização de planos de trabalhos coincidentes, deve-se relacionar no mínimo dois pontos. A origem e o último *pixel* na direção *x* ou *y*.

4 – RESULTADOS

Os resultados foram obtidos a partir da segmentação pelo Método de Otsu. Este método não depende da intervenção humana e nem da forma da distribuição do histograma, tornando o sistema proposto mais independente.

4.1 – Teste de variação de contraste e posição

Este teste foi realizado em nível de simulação, não sendo transmitidas as coordenadas para o controle do robô. Cada objeto foi posicionado de maneira diferente nos três fundos com diferença de tonalidade. Este teste tem o objetivo de verificar a flexibilidade do sistema em relação à mudança de contraste e, além disso, verificar se o contorno obtido está de acordo com o contorno do objeto. O mesmo teste foi realizado com objetos diferentes e os resultados de cada objeto são mostrados separadamente.

As Figuras 4.1 (a), (b) e (c) mostram um objeto posicionado em três fundos de diferentes tonalidades.

As Figuras 4.1 (e) e (f) mostram o histograma do objeto em um fundo pardo e preto, respectivamente. Nota-se uma separação entre o grupo de *pixels* que forma o fundo e o grupo que forma o objeto. A mesma separação não pode ser observada quando o fundo do objeto é branco, Figura 4.1 (d).

O sucesso do processo de segmentação está relacionado com a separação dos grupos de *pixels*. Observa-se nas Figuras 4.1 (g), (h) e (i) o resultado da segmentação. As Figuras 4.1 (h) e 4.2 (i) mostram que o objeto era mais claro em relação ao fundo, diferentemente da Figura 4.1 (g). Um resultado não aceitável é mostrado na Figura 4.1 (g). Neste caso, a sombra do objeto escureceu os *pixels* do fundo, deixando-os com tons de cinza semelhantes ao do objeto. Os *pixels* em torno do seu contorno foram interpretados de maneira incorreta pelo processo de segmentação.

O contorno é obtido através de algoritmos morfológicos que tentam minimizar o efeito de formações de regiões distintas, separando o objeto. Esse efeito é observado na Figura 4.1 (h), onde nota-se uma leve separação do conjunto de *pixels* que formam o objeto. O contorno obtido é mostrado nas Figuras 4.1 (j), (k) e (l).

Ampliando as Figuras 4.1 (j), (k) e (l), nota-se que o erro de contorno gerado pela “falha” do processo de segmentação na Figura 4.1 (m) e o sucesso do processo de segmentação nas Figuras 4.1 (n) e (o).

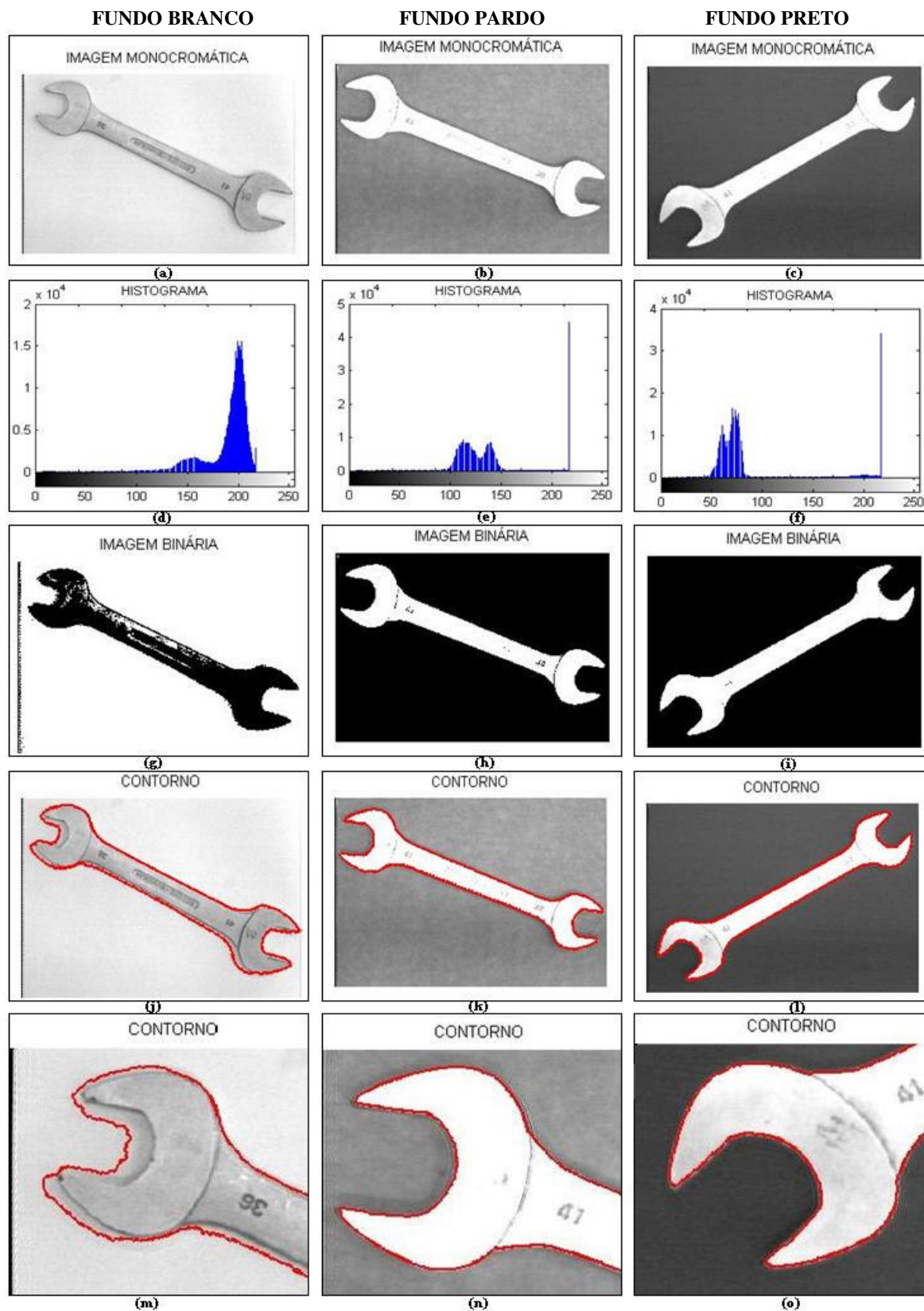


Figura 4.1 – Objetos e seus respectivos histogramas respectivamente (a) e (d), (b) e (e), (c) e (f). Segmentação. Imagem binária com o fundo (g) branco, (h) pardo e (i) preto. O contorno obtido com o fundo (j) branco, (k) pardo e (l) preto. Figuras ampliadas (m), (n) e (o).

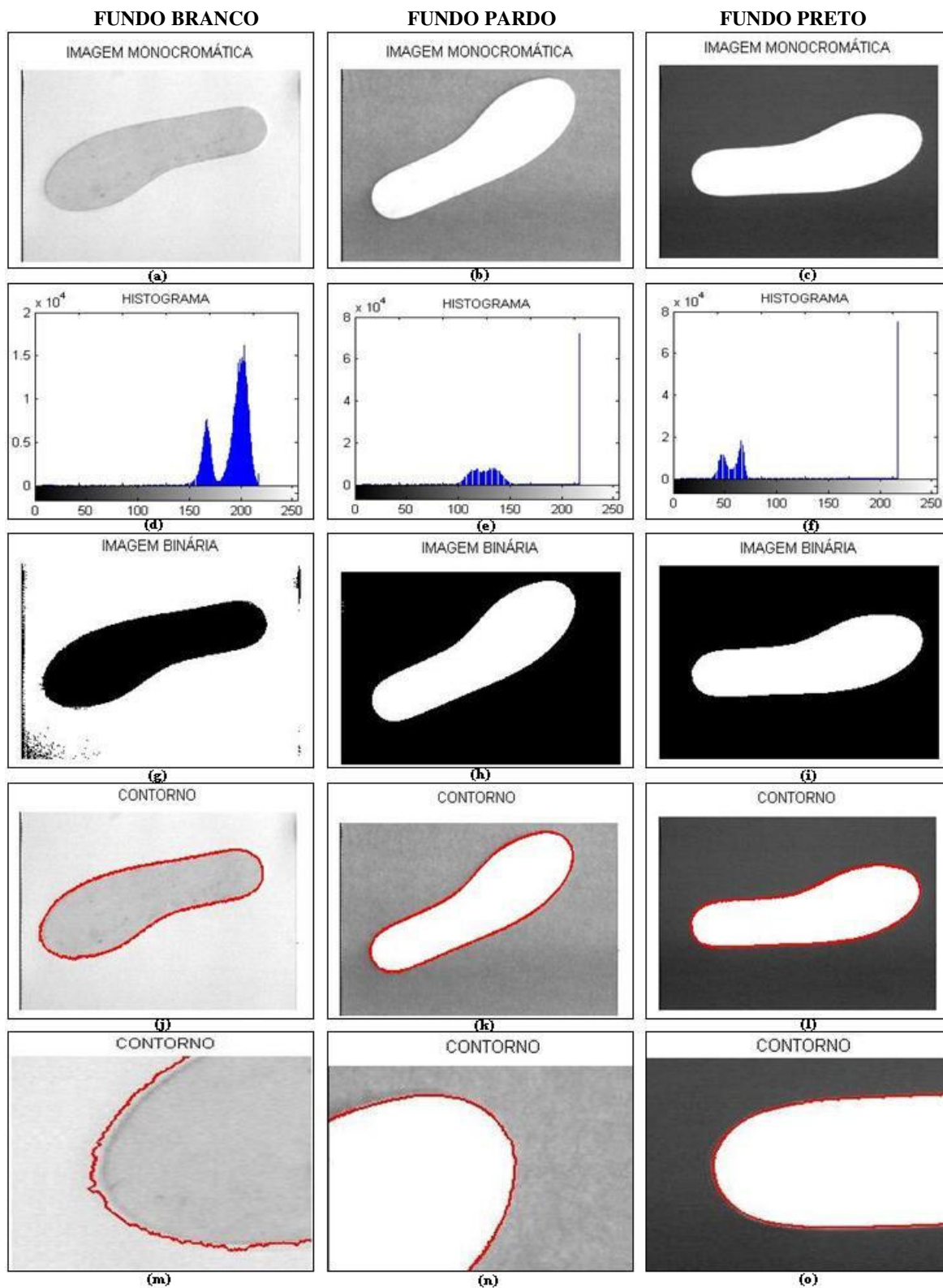


Figura 4.2 – Objetos e seus respectivos histogramas respectivamente (a) e (d), (b) e (e), (c) e (f). Segmentação. Imagem binária com o fundo (g) branco, (h) pardo e (c) preto. O contorno obtido com o fundo (j) branco, (k) pardo e (l) preto. Figuras ampliadas (m), (n) e (o).

A Figura 4.2 mostra outro objeto sendo testado em posições e fundos diferentes e seus respectivos histogramas.

Observando-se a Figura 4.2 (d), nota-se que há uma maior tendência de separação entre os grupos de *pixels*, o que não é observado na Figura 4.1 (d). As Figuras 4.2 (g), (h) e (i) mostram o resultado do processo de segmentação. A Figura 4.2 (g) mostra regiões indesejadas sendo identificadas como pertencentes ao objeto. Tenta-se eliminar essas regiões com algoritmos morfológicos. Caso não seja possível, é estabelecido um tamanho mínimo do conjunto de *pixels* para ser identificado como objeto.

As Figuras 4.2 (j), (k) e (l) mostram o contorno obtido após o processo de segmentação. Nota-se que na Figura 4.2 (m), o erro de contorno não é tão grande quanto na Figura 4.1 (m). Isso se deve pela tendência de formação visível de dois grupos de *pixels* na Figura 4.2 (d).

4.2 – Teste de envio de trajetórias

Este teste tem por finalidade verificar o desempenho do sistema utilizando câmera e resoluções diferentes. A utilização de uma resolução menor diminui a quantidade de pontos tornando a operação mais rápida, porém, a precisão dos pontos para alguns casos pode não ser satisfatória.

Para efeito de testes, na extremidade do robô foi adaptado um dispositivo de fixação para uma caneta, tornando possível visualizar a trajetória seguida pelo manipulador.

As trajetórias podem ser enviadas via *IRBCom* ou carregadas manualmente via disquete (em testes). Nas trajetórias enviadas via *IRBCom*, observa-se a existência de um ciclo que inicia com o envio automático da coordenada do *Boundary Lines* para o robô. Enquanto o robô executa uma trajetória, seu controle permanece bloqueado para comunicação externa. Após término da execução da trajetória, o controle do robô envia um sinal para o *IRBCom* informando que está disponível para executar outra tarefa, reiniciando o ciclo. Considerando a velocidade linear de avanço do robô de 250 mm/s, observa-se um tempo médio de execução total do ciclo de 1,60 s para cada ponto. Nas trajetórias carregadas via disquete, unicamente para efeito de teste, observa-se um tempo médio de 0,15 s por ponto.

A Figura 4.3 mostra a variação de trajetória quando se utiliza câmera de resoluções diferentes. A câmera CCD utilizada, a *SAMSUNG BW-2302ED*, possui uma resolução de 720x480 *pixels*. A *webcam Genius VideoCAM Express V2* possui resolução de 352x288 *pixels*. As Figuras 4.4, 4.5 e 4.6 mostram a trajetória sendo executada e posteriormente comparadas com o objeto.

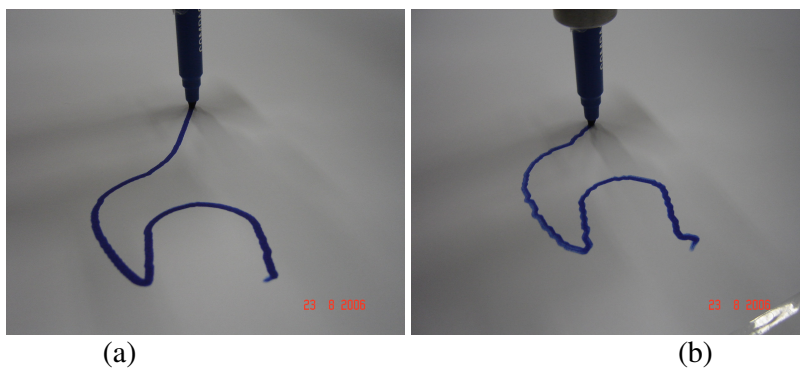


Figura 4.3 – Trajetória obtida através da (a) câmera *CCD* e da (b) *webcam*.

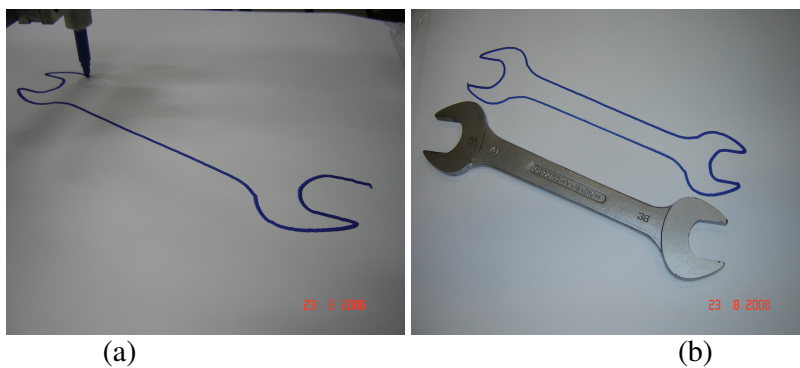


Figura 4.4 – Trajetória obtida da câmera *CCD* (a) e comparação da trajetória com o objeto (b).

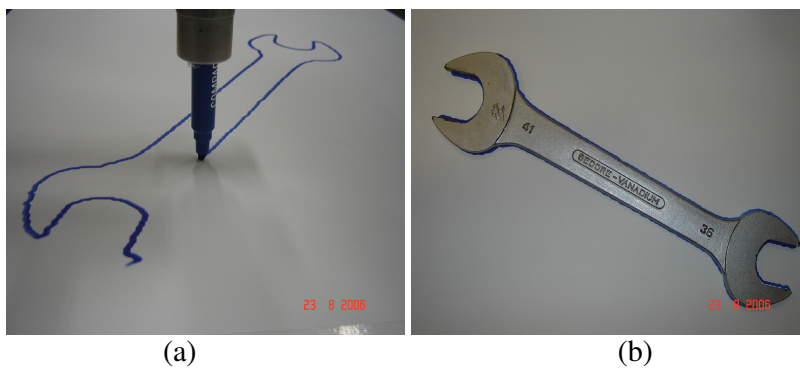
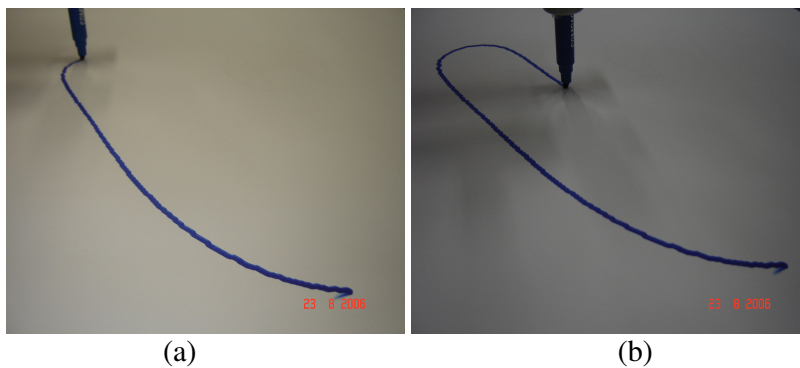


Figura 4.5 – Trajetória obtida da *webcam* (a) e comparação da trajetória com o objeto (b).



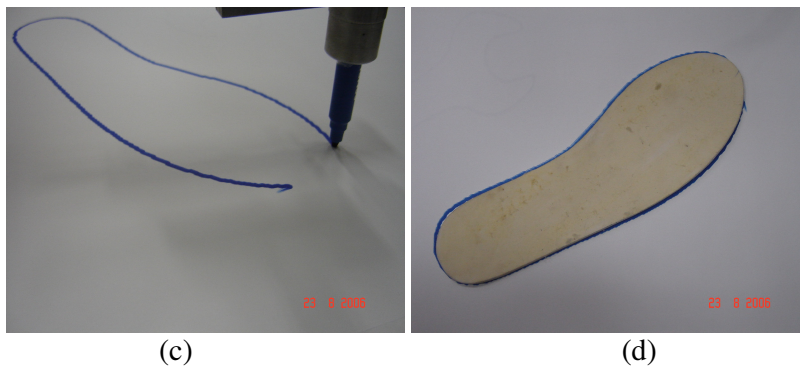


Figura 4.6 – Trajetória obtida da *webcam* (a, b, c) e comparação da trajetória com o objeto (d).

5 – CONCLUSÕES

O Sistema de Visão proposto atende aos objetivos deste trabalho. O sistema se demonstrou flexível em relação às alterações de contraste entre o fundo e o objeto. Com exceção da calibração inicial, o sistema demonstrou autonomia e independência de ajustes ou intervenção humana durante o ciclo de trabalho. O erro de contorno apresentado nas Figuras 4.4 (a) e 4.8 (a) poderiam ser removidos alterando-se a posição da iluminação, visto que o erro foi causado pela sombra do objeto. O uso da câmera CCD tem a vantagem de preservar o sinal analógico da imagem. Para a *webcam*, tem-se a vantagem de não necessitar de uma placa de aquisição de imagens. Modelos de *webcam* com resoluções semelhantes à câmera CCD, poderiam eliminar o problema das trajetórias mostrado na Figura 4.9 (b), justificando-se sua viabilidade. O envio das trajetórias via *IRBCom* é relativamente lento, possuindo um ciclo de operação de aproximadamente 1,5 s para cada ponto, tornando-se característica crítica do sistema. Esse resultado já era esperado com o *IRBCom*, que obteve resultados excelentes em apenas operações *pick-and-place* (Bayer, 2004). Uma solução para maiores taxas de transmissão seria possível via conexão serial ou barramento de rede, opções não disponíveis para o controle do robô utilizado.

O uso do *MATLAB* reduz o tempo de programação, testes e ajustes, permitindo confiabilidade nos resultados. O *Boundary Lines* também mostrou-se uma solução confiável no controle das rotinas e resultados do *MATLAB* e no controle do *IRBCom*.

Para este trabalho, o Sistema de Visão foi projetado para enviar todos os pontos do contorno de um objeto, formados por *pixels* em uma imagem. Uma vez que o sistema obtém a fronteira do objeto independentemente da sua orientação e posição, garante-se que seria de viável implementação no *MATLAB* rotinas para obtenção de outras características geométricas da imagem, tais como área, perímetro, centróide, distâncias e orientações em relação a um determinado referencial.

Para trabalhos futuros, sugere-se:

- Implementar o reconhecimento de outras propriedades geométricas, tais como área, centróide, perímetro, etc.;
- Implementar o uso de trajetórias pré-definidas (retas, parábolas, círculos) sobre os pontos;
- Implementar o reconhecimento e manipulação de objetos em movimento em relação a câmera;
- Implementar um sistema de auto-calibração;

7 – REFERÊNCIAS BIBLIOGRÁFICAS

Albuquerque, Márcio P.; Albuquerque, Marcelo P. **“Processamento de Imagens: Método e Análises”**, Centro Brasileiro de Pesquisas Físicas, Rio de Janeiro.

Alves, J. B. M., 1988 **“Controle de Robô”**, Cartgraf, Campinas, São Paulo. pp 1-23.

Artero, A. O.; Tommaselli, A. M. G., 2002. **“Um Método para a Conexão e Aproximação Poligonal de Elementos de Bordas em Imagens Digitais”**, Universidade Estadual Paulista, Presidente Prudente, São Paulo.

Bayer, F. M., 2004. **“Desenvolvimento e Implementação de uma Interface de Comunicação para Controlador Robótico Industrial”**, Dissertação de Mestrado, Programa de Pós-Graduação em Engenharia Mecânica, UFRGS.

Drobchenko, A.; Et All, 2005. **“Thresholding Based Detection of Fine and Sparse Details”**, Lappeenranta University of Technology, Lappeenranta, Finland

Ferreira, A. P. P., 2003. **“Sistema de Identificação Automática de Matrículas”**, Instituto Superior de Engenharia de Lisboa, Lisboa, Portugal.

Fonseca Neto, J., 1999. **“Aplicação da Transformada de Fourier no Processamento Digital de imagens”**, Aracaju, Sergipe.

Gonzales, R.C.; Woods, R.E., 2000. **“Processamento de Imagens Digitais”**, Edgard Blücher, São Paulo.

Gonzales, R.C.; Woods, R.E.; Eddins, S.L., 2004. **“Digital Image Processing Using MATLAB”**, Pearson Prentice Hall, Upper Saddle River, New Jersey.

Grassi, M. V., 2005. **“Desenvolvimento e Aplicação de um Sistema de Visão Para Robô Industrial de Manipulação”**, Dissertação de Mestrado, Programa de Pós-Graduação em Engenharia Mecânica, UFRGS.

Hirata Nina S. T., 2002. **“Noções de Morfologia Matemática”** Depto. de Ciência da Computação – IME – USP, São Paulo.

Horn, B. K. P., 1986. **“Robot Vision”** McGraw-Hill Book Company, Cambridge, Massachusetts.

Lorini, F. J., 2003. **“Fundamentos de Robótica Industrial”** Notas de aula – Módulo I.

Magrab, E. B., 2000. **“An Engineer’s Guide to MATLAB”**, Prentice Hall, Upper Saddle River, New Jersey.

MATLAB Function Reference v1,v2 e v3, The Math Works

Manozzo, C. E., 2005. **“Desenvolvimento de um Sistema de Visão associado a um Robô Industrial”**, Monografia, Departamento de Graduação em Engenharia Mecânica, UFRGS.

Marchand, P., 1999. **“Graphics and GUIs with MATLAB”** 2° Ed., CRC Press LLC, Florida.

Marques Filho, O.; Vieira Neto, H., 1999. **“Processamento Digital de Imagens”**, Brasport, Rio de Janeiro.

Matsumoto, E.Y., 2004. **“MATLAB 6.5 Fundamentos de Programação”** 2° Ed., Érica Ltda, São Paulo.

McKerrow, P. J., 1990. **“Introduction to Robotics”** Addison-Wesley Publishing Company, Wokingham, England.

Niku, S. B., 2001. **“Introduction to Robotics Analysis, Systems, Applications”**, Prentice Hall, Upper Saddle River, New Jersey.

Shahinpoor, M., 1987. **“A Robot Engineering Textbook”** Haper & Row Publishers, New York. pp. 385-428

Staugaard, A. C., 1987. **“Robotics and AI – An Introduction to Applied Machine Intelligence”** Prentice-Hall, Englewood Cliffs, New Jersey.

Zhang, D.; Lu, G., **“A Comparative Study on Shape Retrieval Using Fourier Descriptors with Different Shape Signatures”**, Monash University, Australia.

<http://www.imageprocessingplace.com/> 15/11/06

<http://www.mathworks.com/applications/imageprocessing/demos.html> 15/11/06

<http://www.ph.tn.tudelft.nl/Courses/FIP/noframes/fip-Contents.html> 15/11/06

http://amath.colorado.edu/courses/4720/2000Spr/Labs/Worksheets/Matlab_tutorial/matlabimpr.html 15/11/06

ANEXO I

Tabela 1 – Especificação do Robô

IRB 1400 Industrial Robot			
Technical Data			
Specification			
Robot	Handling capacity 5 kg	Reach of 5th axis 1.44 m	
Supplementary load	on axis 3 on axis 1	18 kg 19 kg	
Number of axes	Robot manipulator External devices	6 6	
Integrated signal supply	12 signals on upper arm		
Integrated air supply	Max. 8 bar on upper arm		
Performance			
Position repeatability	0.05 mm (average result from ISO test)		
Movements	IRB 1400		
Max. TCP velocity	2.1 m/s		
Continuous rotation of axis 6			
Electrical Connections			
Supply voltage	200–600 V, 50/60 Hz		
Rated power, Transformer rating	4 kVA/7.8 kVA with external axes		
Physical			
Robot mounting	Floor		
Dimensions Robot base	620 x 450 mm		
Weight Robot	225 kg		
Environment			
Ambient temperature Robot unit	5 – 45°C		
Relative humidity	Max. 95%		
Degree of protection	Class D (dry) for welding, machining etc.		
Noise level	Max. 70 dB (A)		
Emmission	EMC/EMI-shielded		
<i>Data and dimensions may be changed without notice.</i>			

Tabela 2 – Especificação da câmera BW-2302EA

Model	BW-2302EA	BW-2302ED	BW-2302CA	BW-2302CH
Power Source	AC24V, 50/60Hz	DC12V	AC24V~, 50Hz	AC230V~, 50Hz
Power Consumption	5W	2W	3.5W	4W
Sync. System	INT/Linelock selectable (Phase:30-300* Power frequency: 60Hz only)	INT Only	INT/Linelock selectable (Phase:30-300* Power frequency: 60Hz only)	INT/Linelock selectable (Phase:30-300* Power frequency: 50Hz only)
Pick up Device	537(H)x505(V) 1/3"CCD		537(H)x597(V) 1/3"CCD	
Scanning System	2:1Interlace(EIA Standard) 525Lines/60Fields/30Frames		2:1 Interlace(CCIR Standard) 625Lines/50 Fields/25Frames	
Minimum Illumination	0.1Lux at F1.2			
Resolution	≥ 380(H)TV Lines			
Video Output	1.0Vp-p EIA comp			
Backlight Compensation	ON/OFF Selectable			
Iris Mode	VIDEO/ESC Selectable			
Electronic Iris	1/60-1/100,000sec		1/50-1/100,000sec	
Lens	C/CS MOUNT Changeable		C/CS MOUNT Changeable	
Gamma Correction	0.7			
AGC	Built-in 30dB			
Aperture Correction	Built-in			
S/N Ratio	≥ 50dB(AGC OFF)			
Ambient Temperature	-10°C+50°C			
Dimension	55(W)x50(H)x136(L)mm(Without Lens)			
Weight	440g(Without Luns)	250g(Without Luns)	440g(Without Luns)	500g(Without Luns)

APÊNDICE A

```

%%%
VERSION:1
LANGUAGE:ENGLISH
%%%
MODULE irbcom2
CONST robtarget P1:=[740,-462.8,739.0],[0.01037,0.26197,0.96473,0.02385],[-1,0,-1,0],[9E+09,9E+09,9E+09,
9E+09,9E+09,9E+09]];
CONST robtarget P2:=[883.89,-197.32,1119.03],[0.027117,0.043878,0.998571,0.014004],[-1,0,-1,0],[9E+09,
9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST num cmdRcvInteger:= 15;
CONST num cmdRcvReal:= 13;
CONST num cmdSetSpeed:= 23;
CONST num cmdDoHome:= 24;
CONST num cmdMoveXY:=10;
var speeddata veloc;
var zonedata zona;
var num posx;
var num posy;
var num command;
var num datavalue;
var bool erro;
var num commandout;
PROC main()
Command:= 0;
! CommandOut: Must be set to the value of the command to send to the PC
CommandOut:= 0;
setgo go1, 0;
setdo do16, 0;
ERRO:= false;
WHILE ERRO=FALSE DO
readcommand;
test command
    case cmddohome:
        dohome;
    case CmdSetSpeed:
        SetSpeed;
    case CmdMoveXY:
        Move;
endtest
if erro=true then
    TPWrite " erro!!! ";
    erro:= false;
endif
ENDWHILE
ENDPROC
PROC DoHome ()
TPWrite "Posicione na Referência";
MoveJ P2,v500,fine,tool0;
ENDPROC
PROC SetSpeed ()
ReadReal;
veloc.v_tep:= DataValue;
ENDPROC
PROC SendInteger ()
!Read first byte
ReadCommand;
DataValue:= Command;
!Read next byte
ReadCommand;
DataValue:= DataValue + (Command * 256);
IF DataValue > 32768 THEN
    DataValue:= 32768 - DataValue;
ENDIF
ENDPROC
PROC ReadReal ()
ReadCommand;
DataValue:= (Command * 256);
!TPWrite "Byte lido= "\Num:= Command;
!Read next byte
ReadCommand;

```



```

DataValue:= (DataValue + Command) * 256;
!TPWrite "Byte lido= "\Num:= Command;
!Read next byte
ReadCommand;
DataValue:= DataValue + Command;
!TPWrite "Byte lido= "\Num:= Command;
IF DataValue > 8388608 THEN
  DataValue:= (8388608 - DataValue) / 256;
ELSE
  DataValue:= (DataValue)/256;
ENDIF
ENDPROC
PROC ReadInteger ()
!Read first byte
ReadCommand;
DataValue:= Command*256;
!Read next byte
ReadCommand;
DataValue:= DataValue + Command;
IF DataValue > 32768 THEN
  DataValue:= 32768 - DataValue;
ENDIF
!TPWrite ". int ="\Num:= DataValue;
ENDPROC
PROC ReadCommand ()
! set "Ready to Receive"
setdo do16, 1;
! wait "PC ok!"
waitDI di16, 1;
! read data, convert bin => decimal
Command:= gi1;
! set "Data Received"
setdo do16, 0;
! wait "PC ok!"
waitDI di16, 0;
ENDPROC
PROC SendCommand ()
IF CommandOut < 16 THEN
! Put data on the output
SetGO go1, CommandOut;
! wait "PC ok!"
if di16<> 1 then
  waitDI di16, 1;
endif
! set "Data out ok!"
setdo do16, 1;
! wait "PC ok!"
waitDI di16, 0;
! set "Data sent"
setdo do16, 0;
waittime 0.01;
ELSE
  ERRO:= TRUE;
ENDIF
ENDPROC
PROC Move()
reset do7;
reset do8;
zona:=z50;
!ler coord x
ReadReal;
posx:=DataValue;
!ler coord y
ReadReal;
posy:=DataValue;
MoveJ Offs(P1,posx,posy,0),v500,fine,tool0;
ENDPROC
ENDMODULE

```

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)