

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
DEPARTAMENTO DE ENGENHARIA MECÂNICA

*Desenvolvimento de um Sistema de Visão para Robô Industrial aplicado na
manipulação de objetos reconhecidos em esteiras transportadoras em movimento*

por

Nilson Cardoso de Lemos

Monografia apresentada ao Departamento
de Engenharia Mecânica da Escola de
Engenharia da Universidade Federal do Rio
Grande do Sul, como parte dos requisitos
para obtenção do diploma de Engenheiro
Mecânico.

Porto Alegre, novembro de 2007.



Universidade Federal do Rio Grande do Sul
Escola de Engenharia
Departamento de Engenharia Mecânica

*Desenvolvimento de um Sistema de Visão para Robô Industrial aplicado na
manipulação de objetos reconhecidos em esteiras transportadoras em movimento*

por

Nilson Cardoso de Lemos

ESTA MONOGRAFIA FOI JULGADA ADEQUADA COMO PARTE DOS
REQUISITOS PARA A OBTENÇÃO DO TÍTULO DE
ENGENHEIRO MECÂNICO
APROVADA EM SUA FORMA FINAL PELA BANCA EXAMINADORA DO
DEPARTAMENTO DE ENGENHARIA MECÂNICA

Prof. Gilberto Dias da Cunha
Coordenador do Curso de Engenharia Mecânica

Área de Concentração: **Projeto e Fabricação**

Orientador: Prof. Flavio José Lorini

Comissão de Avaliação:

Prof. Ângela Rossana Ungaretti

Prof. Eduardo André Perondi

Porto Alegre, novembro de 2007.

ÍNDICE

	Pág.
1. INTRODUÇÃO.....	01
2. REVISÃO BIBLIOGRÁFICA.....	01
2.1. Robô Industrial.....	01
2.2. Visão Robótica.....	02
2.2.1. Imagem.....	03
2.3. Captação da Imagem.....	03
2.3.1. Câmeras digitais.....	04
2.4. Representação de Imagens em um Monitor.....	04
2.5. Técnicas de Iluminação.....	05
2.6. Pré-Processamento.....	05
2.6.1. Domínio espaço.....	06
2.7. Processamento de Imagens.....	06
2.7.1. Monocromatização de Imagens.....	06
2.7.2. Limiarização de Imagens.....	07
2.8. Análise de Imagens.....	08
2.8.1. Momentos.....	08
2.9. Interface de Comunicação PC/Controlador do Robô.....	09
3. DESENVOLVIMENTO E IMPLEMENTAÇÃO.....	09
3.1. Programa de Processamento de Imagens.....	10
3.1.1. Módulo de Captura.....	10
3.1.2. Módulo de Processamento.....	11
3.1.3. Algoritmos e Cálculos.....	14
3.2. Programa de Manipulação.....	17
3.3. Dispositivo de Captura de Imagens.....	17
3.4. Iluminação.....	18
3.5. Módulo IRBCom.....	18
4. AVALIAÇÃO EXPERIMENTAL.....	19
5. RESULTADOS.....	19
6. CONCLUSÃO.....	20
REFERÊNCIAS BIBLIOGRÁFICAS.....	21
APÊNDICE A.....	22
APÊNDICE B.....	23
APÊNDICE C.....	24

1. INTRODUÇÃO

Com o crescimento da população e da competitividade econômica entre as empresas, diversos produtos passaram a ter uma fabricação massiva com a finalidade de suprir a demanda do mercado. Nesse contexto, impõe-se que o setor industrial produza cada vez mais rápido, com mais qualidade e com menor custo. A utilização da robótica industrial, iniciada na década de 1960, contribuiu acentuadamente para a automação em cadeias produtivas. Desde então, a aplicação da robótica cresce a cada dia e acompanha o aumento exponencial da utilização de componentes eletrônicos nos mais diversos setores da indústria.

A utilização de robôs garante uma série de vantagens em relação ao uso de operadores humanos, como o aumento de produtividade e eficiência; a dispensa de utilização de condições de conforto para iluminação, temperatura e ruídos; a ausência de fadiga ou cansaço; a repetibilidade de precisão durante todo o processo. Contudo, os robôs também apresentam desvantagens: não possuem inteligência para reagir de maneira correta em situações não previstas na sua programação; são equipamentos caros e necessitam de mão-de-obra qualificada para manutenção e programação; são limitados em termos de graus de liberdade, resposta em tempo real e sensoreamento do ambiente.

Neste contexto, o presente trabalho visa desenvolver uma ferramenta para atenuar a última desvantagem descrita (limitação em relação à capacidade de sensoreamento do ambiente) através do desenvolvimento de um sistema de visão cujo objetivo é reconhecer diferentes objetos em posições aleatórias no campo de cobertura de uma câmera. Após definidas, as coordenadas de posição são transmitidas ao controle do robô para gerar a trajetória necessária à manipulação dos objetos reconhecidos. O programa aplica-se na visualização e determinação da posição final de um objeto, em um ponto de uma esteira em movimento, e na manipulação que ocorre fora do campo de visão, mas dentro de uma porção do volume de trabalho do robô.

No experimento foram utilizados equipamentos de captura de imagem integrados a um computador interligado ao controle do robô. O sistema (composto pelos módulos de captura e de processamento) foi desenvolvido em linguagem *Visual Basic 2005* e *Visual Basic 6*, utilizando rotinas de transmissão de dados já disponíveis no Laboratório de Robótica do DEMEC e seu funcionamento foi testado nas dependências do mesmo laboratório utilizando o robô ABB IRB1400.

2. REVISÃO BIBLIOGRÁFICA

Dentre as inúmeras técnicas de visão robótica industrial, nesse trabalho são utilizadas apenas as mais comuns e com maior facilidade de implementação. São abordados no texto tópicos relacionados à definição e à descrição dos robôs industriais, introdução à visão industrial e equipamentos utilizados e técnicas de processamento e análise de imagens.

2.1. Robô Industrial

Para definir um robô industrial, pode-se utilizar alguns conceitos internacionais recentes:

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

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

Conforme descrito por Grassi, 2005, um robô industrial é composto, basicamente, de um manipulador, um efetuador, atuadores, sensores, uma unidade de controle principal, uma unidade de controle do robô e dos softwares.

O manipulador é o corpo principal do robô e se constitui por elos, juntas e outros elementos estruturais. Dependendo da configuração desses elementos, o manipulador adquire uma determinada capacidade de movimentos, resultando assim em um volume de trabalho específico. Os robôs são classificados pela configuração das juntas, que são R (Rotacionais) ou P (Prismáticas), ou ainda, de acordo com o seu volume de trabalho resultante.

Considerando-se apenas as juntas principais, os robôs podem ser classificados em Cartesianos ou Retangulares (PPP), Cilíndricos (RPP), Esféricos (RRP), Antropomorfos ou Articulados (RRR) e Scara (RRP).

O efetuador constitui a ferramenta que executa as tarefas específicas como: manipulação, solda, pintura, etc. A possibilidade de acoplamento de variadas ferramentas a um mesmo robô demonstra sua grande versatilidade. O efetuador é comandado pelo controlador do robô ou por alguns dispositivos de controle específico externo.

Os atuadores são responsáveis pela movimentação do robô. Os atuadores mais utilizados são os servo-motores, os motores de passo e os cilindros pneumáticos ou hidráulicos. Os atuadores são comandados pelo controlador do robô.

Os sensores são utilizados para controle de parâmetros ou coleta de dados sobre o estado interno do robô, ou ainda para comunicação com o ambiente externo. Sensores integrados enviam informações sobre cada junta ou link ao controlador, que, por sua vez, determina a configuração do robô. Os robôs também podem ser equipados com sensores externos, que os permitem interagir com o ambiente, possibilitando assim a execução de tarefas mais complexas.

A unidade de controle principal é constituída pela memória de controle e pelo controlador principal. Este controlador recebe dados da memória para controlar os movimentos dos atuadores e coordenar esses movimentos com as informações recebidas dos sensores.

A unidade de controle do robô é constituída pelo processador I/O e pelo processador de controle dos eixos. O processador I/O coordena a entrada e saída de sinais vindos de sensores ou de programas de processamento de dados externos. O processador de controle dos eixos calcula os movimentos das juntas do robô, determinando o quanto e quão rápido uma junta deve mover-se para alcançar uma determinada posição e velocidade. Além disso, supervisiona as ações coordenadas do controlador principal e dos sensores.

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

2.2 Visão Robótica

Um sistema de visão robótica pode ser definido como o processo de extrair, caracterizar e interpretar informações de imagens de um mundo tridimensional. Este processo, também conhecido como visão de máquina ou computador, pode ser subdividido em seis áreas principais: sensoreamento, pré-processamento, segmentação, descrição, reconhecimento e interpretação [Fu et alli, 1987].

“Sensoreamento é o processo de captar a imagem. Pré-processamento trata as imagens com técnicas como redução de ruído e melhoramento de detalhes. Segmentação é o processo que divide uma imagem em objetos de interesse. Descrição lida com a computação de detalhes (tamanho, forma) conveniente para diferenciar um tipo de objeto de outro. Reconhecimento é o processo que identifica esses objetos (chaves, parafusos, blocos de motor). A interpretação dá um significado para os objetos reconhecidos” [Grassi, 2005].

Um sistema básico de visão deve desempenhar duas funções distintas: o processamento e a análise. Processamento da imagem é a soma de rotinas e técnicas que melhoram, simplificam, realçam ou alteram uma imagem. Análise da imagem é a soma de processos nos quais uma imagem que foi preparada por processamento é analisada para extrair informação, por exemplo, identificar objetos ou fatos sobre o objeto ou seu ambiente.

Os sistemas de visão robótica mais atuais são compostos por câmeras de alta resolução e de rotinas de processamento que incluem momentos estatísticos, coeficientes da transformada de *Fourier* e medidas de distâncias multidimensionais. Além disso, há estudos sobre sistemas de visão que utilizam redes neurais de base radial para a análise das imagens [Simões, 2005].

Para um melhor entendimento do presente trabalho e compreensão da metodologia adotada, faz-se necessária a revisão de alguns conceitos e equipamentos.

2.2.1 Imagem

Imagem é a representação de uma cena real, colorida ou preto-e-branco, impressa ou digital. O termo *imagem monocromática*, ou *em escala de cinza*, conforme mostra a Figura 2.1, refere-se à função bidimensional de intensidade da luz $f(x,y)$, onde x e y denotam as coordenadas espaciais (sendo x o eixo horizontal e y o eixo vertical, com origem no canto superior esquerdo da imagem), e o valor de f em qualquer ponto (x,y) é proporcional ao brilho (ou níveis de cinza) da imagem naquele ponto.

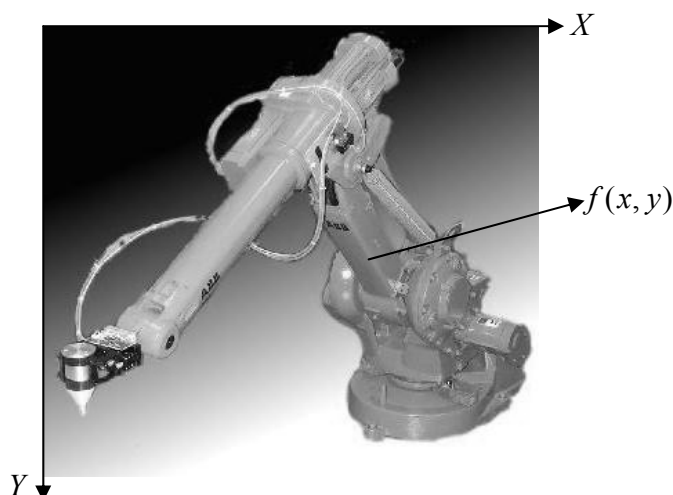


Figura 2.1 – Imagem monocromática.

Ou seja, uma imagem monocromática é uma função $f(x,y)$ discretizada, tanto em coordenadas espaciais quanto em brilho. Uma imagem digital pode ser considerada uma matriz cujos índices de linhas e de colunas identificam um ponto na imagem. O correspondente valor do elemento dessa matriz identifica o nível de cinza naquele ponto. Os elementos dessa matriz digital são chamados de elemento de imagem, elementos de figura ou pixels, sendo este último abreviatura de “*picture element*” [Gonzales, 2003].

2.3 Captação da Imagem

Dois elementos são necessários para a aquisição de imagens digitais, conforme Gonzales, 2003. O primeiro é um dispositivo físico que seja sensível a uma banda do espectro de energia eletromagnética (como raios-x, ultravioleta, visível, ou banda infravermelha) e que produza um sinal elétrico de saída proporcional a um nível de energia percebida. O segundo, chamado digitalizador, é um dispositivo para a conversão da saída elétrica de um dispositivo de sensoreamento físico para a forma digital.

A captação de imagens para um sistema de visão robótica é realizada através de câmeras que podem ser analógicas ou digitais. Câmeras analógicas não são as mais comuns, mas ainda são aplicadas, já as câmeras digitais são mais comuns e são similares entre elas. A forma de captura pode ser analógica ou digital, mas, em sistemas de visão, a imagem necessita ser digitalizada. Na forma digital, todos os dados são binários e armazenados em um arquivo digital.

O funcionamento das câmeras vidicon, segundo Gonzales, 2003, é baseado no princípio da fotocondutividade. Esse é um tipo de câmera analógica, que transforma a imagem em um sinal analógico: um padrão de condutividade variável é produzido na superfície de um tubo por uma imagem focalizada e esse padrão é acompanhado pela distribuição de brilho na imagem óptica. Um feixe de elétrons independente e precisamente focalizado trilha a parte de trás da superfície do alvo fotocondutivo e, por meio da neutralização de carga, o feixe cria uma diferença de potencial que gera em um coletor um sinal proporcional ao padrão de brilho da

entrada. Uma imagem digital pode ser obtida ao se quantizar esse sinal e a posição correspondente do feixe.

2.3.1 Câmera Digital

As câmeras digitais são baseadas na denominada “*tecnologia do estado sólido*”. Como em outras câmeras, um conjunto de lentes é usado para projetar a área de interesse sobre a área da imagem de interesse da câmera. A maior parte da câmera é uma área de imagem de silício em estado sólido que possui centenas de milhares de pequenas áreas fotossensíveis chamadas *photosites*, conforme a Figura 2.2. Cada pequena área é um píxel. Assim que a imagem é projetada sobre a área da imagem, em cada píxel uma carga é criada proporcional à intensidade da luz em cada ponto (uma câmera digital é também chamada de dispositivo de carga acoplada ou câmera *CCD* e dispositivo de carga integrada ou câmera *CID*). O conjunto de cargas, lido seqüencialmente, pode ser uma representação dos píxels de uma imagem monocromática [Okigawa, 1992].

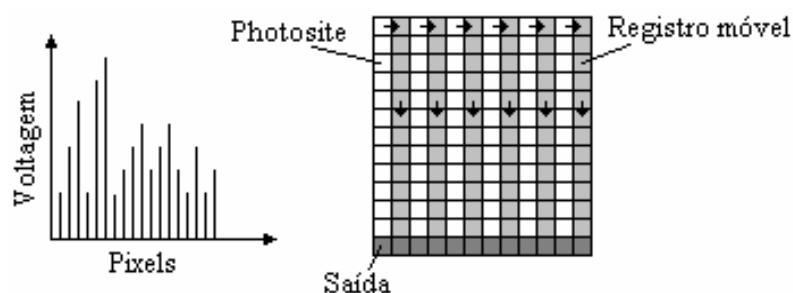


Figura 2.2 – Modelo de captura de dados de imagem.

2.4 Representação de imagens em um monitor

Para criar imagens em movimento em televisões ou monitores, a imagem é varrida e reconstruída um determinado número vezes por segundo, dependendo do padrão utilizado. Para os olhos humanos, que possuem uma histerese temporária de cerca de 1/10 de segundo, imagens mudando mais de 10 vezes por segundo são percebidas como contínuas e em movimento.

As *TV's* e os monitores de vídeo têm sua tela baseada em um tubo, o *CRT* (Tubo de Raios Catódicos). A tela deste tubo é composta por camadas de fósforo, que são atingidas por elétrons através de "disparos" feitos por um canhão localizado no início do tubo. Quando o elétron encontra o fósforo, uma luz é gerada naquele ponto e é esse efeito que faz com que a imagem apareça na tela do monitor. Para gerar as imagens, o canhão percorre toda a extensão da tela, ponto por ponto, linha por linha. Como cada ponto de luz tem duração curta e a imagem precisa ser constantemente atualizada, esse processo, conhecido como varredura, é repetido a todo instante.

Para imagens coloridas, a imagem projetada é decomposta em três cores, vermelho, verde e azul (*RGB*, abreviatura de *Red, Green, Blue*). O processo de varredura é repetido para as três imagens e três sinais simultâneos são produzidos e transmitidos. No monitor, três canhões de elétrons regeneram três imagens simultâneas em *RGB* na tela, exceto quando a tela possui três conjuntos de pequenos pontos (sub-píxels) que reagem fosforescendo em cores *RGB* e são repetidos sobre a tela inteira. Cada conjunto de três pontos, sendo um vermelho, um verde e um azul, é denominado tríade, ou píxel, conforme a Figura 2.3. *Dot Pitch* é a distância entre dois pontos da mesma cor. Quanto menor esta distância maior será a qualidade da imagem [InfoWester].

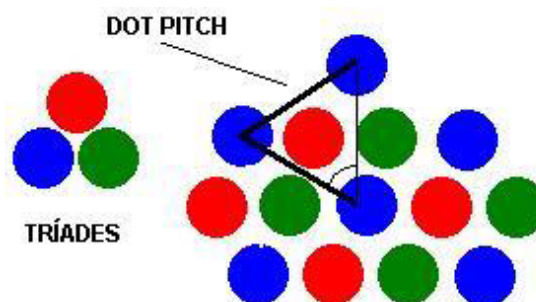


Figura 2.3 – Triáde RGB e *Dot Pitch*.

Os canhões de elétrons recebem um sinal que indica qual tom, de sua respectiva cor, que cada sub-píxel da *triáde RGB* assume. Um monitor típico do ano de 2007 utiliza até 24 bits de informação para cada píxel. Geralmente, a partição é de oito bits para cada cor (vermelho, verde e azul), dando um alcance de 256 possíveis valores, ou intensidades, para cada tom. Por usar uma combinação apropriada para as intensidades de vermelho, verde e azul, diversas cores podem ser representadas. Com este sistema, mais de 16 milhões (16.777.216 ou 256^3) de diferentes combinações de tons podem ser especificadas, mesmo que não sejam facilmente distinguidas.

As imagens coloridas, nesse sistema, são divididas em imagens *RGB* e tratadas como três imagens separadas. Quando o sinal não precisa ser transmitido, ele pode ser gravado para uso posterior, digitalizado ou inserido em um monitor para visualização direta.

2.5 Técnicas de Iluminação

Iluminação de uma cena é um fator importante que freqüentemente afeta a complexidade de algoritmos de visão. Iluminação arbitrária do ambiente freqüentemente é inaceitável porque pode resultar em imagens com pouco contraste, reflexões, sombras e detalhes estranhos. Um sistema de iluminação bem projetado ilumina a cena de tal modo que a complexidade da imagem adquirida é minimizada, enquanto a informação requerida para a detecção de objetos e sua extração é melhorada [Grassi, 2005].

As técnicas de iluminação mais comuns são: a direta, a difusa, a traseira, a estruturada e a direcional. Na iluminação direta, uma fonte de luz é disposta diretamente acima do objeto. A iluminação difusa pode ser empregada para objetos lisos com superfícies regulares. Iluminação traseira produz uma imagem preto-e-branco, ideal para aplicações onde silhuetas de objetos são suficientes para reconhecimento. A iluminação estruturada consiste de projetar pontos, listras ou grades na superfície de trabalho. A iluminação direcional é útil principalmente para inspeção superficial de objetos, onde defeitos na superfície medindo modificam o espalhamento da luz. Em superfícies livres de defeitos há pouco espalhamento.

2.6 Pré-Processamento

Existem diversas técnicas de pré-processamento de imagens passíveis de serem empregados em visão robótica. Os dois principais métodos são os baseados em técnicas de domínio-freqüência e os baseados em técnicas de domínio-espço.

No processo de domínio-freqüência, o espectro de freqüência da imagem é usado para alterar, analisar ou processar a imagem. Neste caso, pixels individuais e seus conteúdos não são usados, sendo uma representação em freqüência da imagem inteira usada no processo. No processo de domínio-espço, o processo é aplicado a pixels individuais na imagem. Como resultado, cada píxel é afetado diretamente pelo processo [Fu et alli, 1987].

As duas técnicas são igualmente importantes e poderosas e são usadas para diferentes propósitos, ressaltando-se que apesar de serem aplicadas diferentemente, são relacionadas. Nesse trabalho são utilizadas as técnicas de domínio-espço devido à facilidade de implementação e ao menor tempo de processamento, descritas em seguida.

2.6.1 Domínio-Espaço

O termo domínio espacial refere-se ao agregado de pixels que compõem uma imagem, e métodos no domínio espacial são procedimentos que operam diretamente sobre estes pixels. Funções de processamento de imagens no domínio espacial podem ser expressas como

$$g(x, y) = h[f(x, y)] \quad (1)$$

onde $f(x, y)$ é a imagem de entrada, $g(x, y)$ é a imagem processada e h é um operador sobre f , definido sobre alguma vizinhança de (x, y) [Gonzales, 2003].

2.7 Processamento de Imagens

As técnicas de processamento da imagem são usadas para realçar, melhorar ou alterar uma imagem e prepará-la para sua análise, eliminando ruídos e elementos indesejados, mas que são inerentes ao processo de captura de imagens. Nenhuma informação é extraída de uma imagem durante o seu processamento.

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

2.7.1 Monocromatização de Imagens

Existem diversos tipos de padrões de cores. Basicamente, o propósito de um modelo é facilitar a especificação das cores em alguma forma padrão. Essencialmente, um modelo de cor é uma especificação de um sistema de coordenadas tridimensionais e um subespaço dentro desse sistema onde cada cor é representada por um único ponto.

Imagens no modelo *RGB* consistem em três planos de imagem independentes, uma para cada cor primária. Quando alimentada num monitor *RGB*, essas três imagens combinam-se sobre a tela fosfórea para produzir uma imagem de cores compostas. Assim, o uso desse modelo para processamento de imagens faz sentido quando as imagens são naturalmente expressas em termos de planos de três cores. A maioria das câmeras coloridas utilizadas para aquisição de imagens digitais e os monitores coloridos utilizam o formato *RGB* [Gonzales, 2003].

No modelo de cores *RGB*, o subespaço de cores de interesse é o cubo unitário mostrado na Figura 2.4, no qual os valores *RGB* estão nos três cantos; ciano, magenta e amarelo estão nos outros três cantos; preto está na origem e branco está no canto mais distante da origem. Nesse modelo, a escala de cinza estende-se do preto até o branco ao longo da linha juntando esses dois pontos, e as cores são pontos sobre ou dentro do cubo, definidas por vetores estendendo-se a partir da origem.

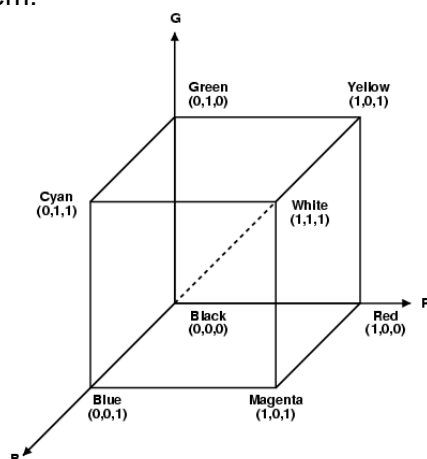


Figura 2.4 – Cubo de Cores *RGB*.

Já o modelo *YIQ* (*luminance*, *in-phase* e *quadrature*) é usado na transmissão comercial de TV colorida. Basicamente, é uma recodificação de *RGB* para eficiência de transmissão e

para manutenção da compatibilidade com os padrões monocromáticos de TV. De fato, o componente Y do sistema fornece todas as informações de vídeo necessárias para um aparelho de TV monocromático. A conversão de RGB para YIQ é definida como

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0,299 & 0,587 & 0,114 \\ 0,596 & -0,275 & -0,321 \\ 0,212 & -0,523 & 0,311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2)$$

Para se obter os valores RGB a partir de um conjunto de valores YIQ , simplesmente realizamos a inversão da matriz. Outra conclusão interessante é que, para transformarmos uma imagem colorida numa imagem em tons de cinza basta isolar a componente Y do sistema YIQ , ou seja

$$Y = 0,299R + 0,587G + 0,114B \quad (3)$$

2.7.2 Limiarização de Imagens

Limiarizar é o processo comparar a intensidade de cor de cada de píxel com um determinado limite e então assinalar se esse nível está abaixo do limite (desligado, zero ou não pertence) ou acima do limite (ligado, 1 ou pertence). A limiarização também pode ser executada tanto para um nível único quanto para vários níveis, nos quais a imagem processada é dividida em camadas, sendo cada uma dessas camadas com um limite selecionado [Grassi, 2005].

Um dos procedimentos de limiarização mais comuns em sistemas de visão utiliza apenas imagens binárias. Para isso, são utilizados apenas objetos claros com fundos escuros, ou o contrário. O objetivo é garantir um alto contraste para que o processo de limiarização seja bem sucedido. Esta condição pode ser alcançada com iluminação controlada em condições industriais. Em imagens binárias, os pixels são “ligados” ou “desligados” e, logo, escolher um limite é simples. Portanto, a limiarização de imagens é uma operação que envolve testes contra uma função T da forma [Gonzalez e Woods, 2003]:

$$T = T[x, y, p(x, y), f(x, y)] \quad (4)$$

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

$$g(x, y) = \begin{cases} 1 & \text{se } f(x, y) > T \\ 0 & \text{se } f(x, y) \leq T \end{cases} \quad (5)$$

Observando $g(x, y)$, constata-se que qualquer píxel com seu nível de cinza abaixo do limite T será rotulado como 0, e, por conseguinte, qualquer píxel com seu nível de cinza acima do limite T será rotulado como 1. Essa seria a forma ideal do processo de limiarização para um objeto escuro sobre um fundo claro. A Figura 2.5 exemplifica esta operação com um objeto genérico e diversas sombras ou elementos indesejados na cena captada. Após a limiarização, a cena fica “limpa”, podendo-se tratar apenas com o objeto.

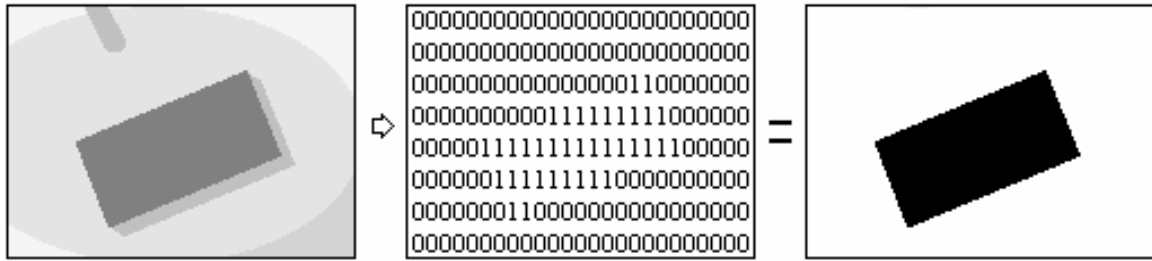


Figura 2.5 – Processo de limiarização.

2.8 Análise de Imagens

A análise de imagens é um conjunto de técnicas e operações que são usadas para extrair informações de imagens. Entre estas operações têm-se as seguintes técnicas de análise: reconhecimento de objetos, extração de características superficiais, determinação de orientação, posição e tamanho e outras propriedades de objetos. Nesse trabalho, é descrito o procedimento para obtenção da área e do centróide de um objeto aleatório, através do método dos momentos.

2.8.1 Momentos

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

$$M_{a,b} = \sum_{x,y} x^a y^b \quad (6)$$

Onde $M_{a,b}$ é o momento do objeto na imagem com índices a e b . As coordenadas de cada píxel “ligado” na imagem são representadas por x e y .

Segundo Gonzales, 2003, uma rotina baseada na Equação (5) deve determinar se cada píxel pertence ao objeto (está “ligado”) e, caso positivo, irá elevar as coordenadas da localização do píxel aos dados valores de a e b . O somatório desta operação sobre toda a imagem consiste no momento particular do objeto nos valores a e b . É comum a aplicação de momentos que usem todas as combinações de índices entre 0 e 3.

As distâncias x e y são medidas tanto dos eixos coordenados fictícios localizados na borda da imagem (x,y) ou são medidas dos eixos coordenados formados pelas primeiras linha e coluna da imagem, conforme a figura 2.6. As distâncias são medidas contando-se o número de pixels do eixo até o píxel de interesse. O uso da primeira linha e coluna como eixos coordenados é mais comum, mas, como ambas as referências são usadas consistentemente, a escolha não é essencial.

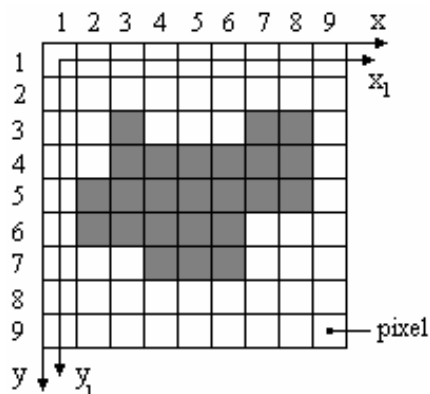


Figura 2.6 – Objeto aleatório e a representação dos seus pixels.

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

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

$$\bar{y} = \frac{\sum y}{\text{área}} = \frac{M_{0,1}}{M_{0,0}} \quad (7)$$

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

$$\bar{x} = \frac{\sum x}{\text{área}} = \frac{M_{1,0}}{M_{0,0}} \quad (8)$$

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

2.9 Interface de comunicação PC/Controlador do Robô

A comunicação entre o PC e o robô se dará através do módulo de comunicação IRBCom. Segundo Bayer, 2004, o robô ABB IRB1400 era, originalmente, desprovido de uma interface de comunicação de dados. Para a implementação de um sistema de transmissão de dados utilizou-se a porta paralela do PC e uma placa E/S Digital para sinais discretos disponível no controlador do robô. Devido à diferença das características elétricas entre as interfaces utilizadas, foi necessário projetar um dispositivo que permitisse o ajuste dos níveis de tensão entre os sinais.

O sistema foi elaborado visando sua utilização por programas desenvolvidos pelos usuários em ambiente *Microsoft Windows*, sendo disponibilizadas rotinas para envio e recebimento de dados através de um protocolo próprio. Na plataforma PC as rotinas estão encapsuladas em um arquivo compilado no formato *DLL (Dynamic Link Library – Biblioteca de Vínculo Dinâmico)*.

O programa desenvolvido pelo usuário é responsável por todo o processamento das informações, que são enviadas através do sistema de comunicação a outro programa específico, que é executado no controlador do robô e desenvolvido na linguagem *ABB RAPID*, o qual interpreta os dados e ativa as tarefas correspondentes.

3. DESENVOLVIMENTO E IMPLEMENTAÇÃO DO PROGRAMA SVR

O foco desse trabalho é o desenvolvimento e a aplicação de um sistema de visão para tarefas de manipulação robótica industrial. Devido a isso, pode ser dividido em cinco módulos principais:

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

Essa divisão torna a execução do projeto mais objetiva. Apesar de serem módulos independentes, operam de forma conjunta para formar o sistema de visão proposto.

3.1 Programa de Processamento de Imagens

O objetivo principal do programa SVR (Sistema de Visão Robótica) é obter as informações necessárias para que o robô possa executar tarefas de manipulação. O SVR é subdividido em dois módulos distintos, denominados “*Captura de Imagem*” e “*Processamento de Imagens*”. Sendo assim, o primeiro captura as imagens através de câmera CCD e as exibe em um monitor RGB, conectado ao PC. O segundo módulo realiza as tarefas de processamento e análise das imagens. Logo após, envia os dados necessários à manipulação para o controlador do robô. Um programa de manipulação aguarda o recebimento dessas informações.

3.1.1 Módulo de captura

O módulo de captura de imagens foi desenvolvido utilizando-se a linguagem de programação *Microsoft Visual Basic 2005*, o qual efetua a captura das imagens obtidas pela câmera utilizada. Sua função básica é a geração de imagens para a operação do módulo de processamento de imagens. Se o módulo de processamento estiver desativado, o módulo de captura serve como uma ferramenta de visualização de imagens, tendo a opção de salvar uma imagem qualquer. A interface do programa é ilustrada na figura 3.1.

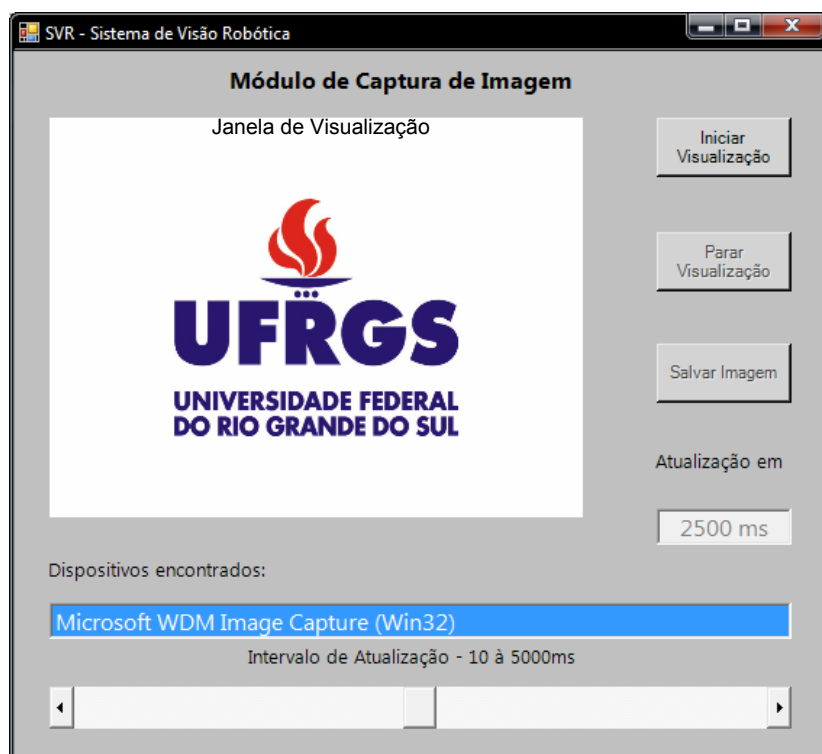


Figura 3.1 – Interface do Módulo de Captura de Imagens.

O módulo de captura, por meio do seu algoritmo, não procura o hardware específico de captura de imagem, mas exibe na janela denominada “*Dispositivos encontrados*” uma lista de todos os drivers de captura de imagens instalados no PC. Portanto, o primeiro passo para iniciar o processo de aquisição de imagens é definir qual dispositivo será utilizado, no caso de um PC possuir mais de um instalado. Se houver apenas um, ele será mostrado na mesma janela, mas sem a barra de rolagem vertical que oferece a possibilidade de seleção.

Em seguida, deve-se selecionar a taxa de atualização da imagem, ajustando a barra de rolagem horizontal denominada “*Intervalo de Atualização*”. Esse parâmetro define em qual período de tempo, em milissegundos, o sistema atualizará a imagem capturada. O tempo mínimo de atualização da imagem é de 10 milissegundos (0,01 segundo) e o tempo máximo é

de 5000 milissegundos (5 segundos). O intervalo de atualização selecionado é mostrado na janela “Atualização em”.

Ao clicar no botão “Iniciar Visualização”, será acionado o dispositivo de captura de imagem. Essa será mostrada na “Janela de Visualização”, sendo renovada a cada intervalo de tempo pré-selecionado. Para interromper o processo basta clicar no botão “Parar Visualização”. Enquanto o procedimento de captura de imagem estiver ativado, é possível capturar uma imagem e salvá-la em um dispositivo ou em um endereço personalizável, clicando no botão “Salvar Imagem”.

É importante salientar que a janela do módulo de captura não pode ser movida de sua posição original no monitor durante o processamento das imagens. Essa limitação será esclarecida no tópico que descreve o funcionamento do algoritmo do módulo de processamento.

3.1.2 Módulo de Processamento

O sub-aplicativo de processamento de imagens foi desenvolvido utilizando-se a linguagem de programação *Microsoft Visual Basic 6*. A mudança da versão do compilador desse módulo é devido à utilização de um comando específico dessa versão.

Nessa etapa, é realizado o processamento e a análise das imagens obtidas pelo primeiro sub-aplicativo. Em seguida, os dados obtidos são enviados ao controlador do robô, onde um programa de manipulação estará aguardando sua chegada. A interface do programa é ilustrada na figura 3.2.

Módulo de Processamento de Imagens													
Ativação do Processamento <input type="button" value="Iniciar Processamento"/> <input type="button" value="Parar Processamento"/>													
Configuração Intensidade: 128 <input type="button" value="Calibrar a Área"/> Intervalo: 10 <input type="button" value="Reset"/> Velocidade: 200 <input type="button" value="Referenciar"/> <input type="button" value="Velocidade"/> 1 Pixel = "x" mm: 1 1 Pixel = "y" mm: 1													
Resultados <table border="1"> <tr> <td>Status do Programa</td> <td>DESATIVADO</td> </tr> <tr> <td>Nº. de Peças Processadas</td> <td>0</td> </tr> <tr> <td>Área da Peça Atual</td> <td>0</td> </tr> <tr> <td>Área Máx. Estimada</td> <td>0</td> </tr> <tr> <td>Área Média Estimada</td> <td>0 + 0</td> </tr> <tr> <td>Área Mín. Estimada</td> <td>0</td> </tr> </table>		Status do Programa	DESATIVADO	Nº. de Peças Processadas	0	Área da Peça Atual	0	Área Máx. Estimada	0	Área Média Estimada	0 + 0	Área Mín. Estimada	0
Status do Programa	DESATIVADO												
Nº. de Peças Processadas	0												
Área da Peça Atual	0												
Área Máx. Estimada	0												
Área Média Estimada	0 + 0												
Área Mín. Estimada	0												
Coordenadas do Centróide <table border="1"> <thead> <tr> <th>Pixels</th> <th>Milímetros</th> </tr> </thead> <tbody> <tr> <td>X: 0</td> <td>X: 0</td> </tr> <tr> <td>Y: 0</td> <td>Y: 0</td> </tr> </tbody> </table>		Pixels	Milímetros	X: 0	X: 0	Y: 0	Y: 0						
Pixels	Milímetros												
X: 0	X: 0												
Y: 0	Y: 0												

Figura 3.2 – Interface do Módulo de Processamento de Imagens.

Esse sub-aplicativo é dividido em quatro partes distintas. Essa divisão organiza a interface e também agrupa procedimentos semelhantes. São elas: configuração, arquivo, ativação do processamento e resultados.

Configuração

As variáveis de configuração do sistema estão reunidas nesse campo. São elas que determinam quais serão os parâmetros utilizados pelo algoritmo para o processamento das imagens.

As variáveis de configuração do sistema estão reunidas nesse campo. São elas que determinam quais serão os parâmetros utilizados pelo algoritmo para o processamento das imagens.

A primeira barra de rolagem horizontal, denominada “*Intensidade*”, define qual será o valor limite de intensidade de cinza que será utilizado como valor-limite para a obtenção da imagem binária. O programa utiliza como padrão o valor 128, mas pode variar entre 1 e 256. O valor selecionado é mostrado na caixa de texto acima da barra de rolagem.

A segunda barra de rolagem horizontal, denominada “*Velocidade*”, determina a velocidade, em mm/s (milímetros por segundo) com que o robô irá se mover para efetuar as tarefas de manipulação. O valor é enviado ao controlador do robô clicando no botão “*Velocidade*”, desde que o programa de manipulação possua uma variável aguardando esses dados. O programa utiliza como padrão o valor mínimo (200 mm/s). Entretanto, pode variar entre 200 e 2000 mm/s. O valor selecionado é mostrado na caixa de texto acima da barra de rolagem.

A terceira barra de rolagem, denominada “*Intervalo*”, determina em qual intervalo de tempo, em segundos, o processamento de imagens será ativado. O valor determinado nesse campo será sempre somado ao tempo de processamento da imagem. O programa é utilizado como padrão o valor 10 segundos, mas pode variar entre 1 e 20 segundos. O valor selecionado é mostrado na caixa de texto acima da barra de rolagem.

As caixas de texto denominadas “*1 pixel = x mm*” e “*1 pixel = y mm*” determinam os fatores de conversão de pixel para milímetros, respectivamente, para os eixos *X* e *Y*. Esses valores são uma função da distância focal, ou seja, da distância entre a superfície e o dispositivo de captura de imagem. O programa calcula o centróide do objeto em pixels, mas o valor aguardado pelo programa de manipulação deve estar em milímetros. Assim, é necessário que um procedimento de calibragem seja efetuado para que esse campo possa ser completado com o valor correto. Os valores estipulados serão utilizados como constantes na relação linear que fará a transformação de pixels para milímetros. O procedimento de calibração é descrito na seção 3.1.3. O programa utiliza como padrão para os dois fatores o valor 1, ou seja, 1 pixel é igual a 1 milímetro.

O botão denominado “*Calibrar Área*” realiza um processo de estimativa da área do objeto a ser calculado. Para isso, coloca-se o objeto sob a câmera e, ao pressionar esse botão, o programa repete a rotina de cálculo da área dez vezes. Após, calcula a área média e as tolerâncias admissíveis. Esse procedimento é importante, pois, como a esteira estará em movimento, o programa utilizará essa informação para determinar se o objeto está totalmente ou parcialmente sob o campo de visão.

O botão denominado “*Referenciar*” indica ao programa de manipulação qual será o ponto de referência. Isto é, as coordenadas da posição dos objetos sob a câmera são calculadas em relação a esse ponto, que é o canto superior esquerdo da janela de captura de imagens. O comando é enviado ao controlador do robô clicando-se nesse botão, desde que o programa de manipulação possua uma variável aguardando esses dados.

O botão “*Reset*” zera todos os contadores e redefine as variáveis de configuração nos seus respectivos valores padrões.

Arquivo

As variáveis de comunicação, de teste e de encerramento do módulo estão reunidas nesse menu. As três funções, conforme a figura 3.3, são: salvar em texto, modo de simulação e sair.

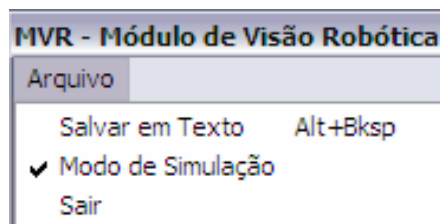


Figura 3.3 – Funções do menu Arquivo.

A opção “*Salvar em Texto*”, quando selecionada, grava um arquivo no formato *Microsoft Notepad* (*.txt) sempre no endereço C:\. Esse arquivo contém uma matriz, onde cada elemento é a representação binária resultante do processamento do pixel na posição correspondente da imagem analisada. Esse arquivo serve, principalmente, para teste e verificação do sistema,

onde pode-se checar se as condições ideais de análise foram atingidas. A figura 3.4 ilustra o arquivo binário resultante do processamento da imagem de fundo da janela de visualização do módulo de captura de imagens. Os espaços escuros são pixels “desligados”, ou zero; os claros, pixels “ligados” ou 1.

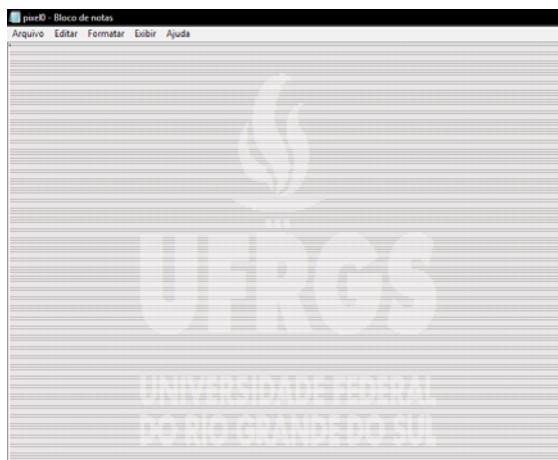


Figura 3.4 – Matriz binária resultante do processamento de uma imagem.

A opção “*Modo de Simulação*”, quando selecionada, não envia comandos ou coordenadas ao controlador do robô. Essa opção é importante para testes e simulações do sistema, pois permite que se possa realizar um ajuste fino do módulo de processamento sem que o robô seja ativado a cada tentativa. A opção “*Sair*” finaliza o programa.

Ativação do Processamento

Nesse campo, encontram-se os botões que ativam e desativam o processamento das imagens. Obviamente, ao clicar no botão “*Iniciar Processamento*”, a rotina de cálculo da área e do centróide do objeto a ser analisado é executada a cada intervalo de tempo pré-determinado. Porém, se a calibragem do objeto não for realizada, o procedimento é interrompido após o primeiro ciclo. Ao clicar no botão “*Desativar Processamento*”, a análise dos objetos é interrompida.

Resultados

Todos os resultados relevantes dos cálculos realizados pelo módulo estão reunidos nesse campo. Alguns servem apenas para informação do operador, enquanto outros são os valores que serão enviados ao programa de manipulação.

O campo denominado “*Status do Programa*” indica se o módulo de está ou não está processando alguma imagem. Se não estiver, a mensagem “*desativado*” é exibida, mas, se houver processamento, um cronômetro progressivo indica o tempo decorrido desde o último objeto analisado, com precisão de décimos de segundo.

O campo denominado “*Nº. de Peças Processadas*” indica quantos objetos o sistema já processou desde que foi ativado. Se o processamento das imagens for interrompido e, em seguida, reativado a contagem continuará no mesmo valor em que parou. Se o botão “*Reset*” for pressionado, o contador é zerado.

O campo denominado “*Área Atual da Peça*” exibe a área, em pixels, do último objeto processado. Já o campo denominado “*Área Média da Peça*” exibe a área média e os seus desvios permitidos obtidos no processo de calibração da imagem do objeto. Os campos “*Área Máx. Estimada*” e “*Área Mín. Estimada*” exibem, respectivamente, os limítar máximo e mínimo permitidos para a área do objeto. Ou seja, cada objeto processado deverá possuir uma área entre esses dois valores e, se isso não ocorrer, o programa exibe uma advertência e é desativado.

Os campos denominados “*Coordenadas do Centróide*” exibem, tanto em pixels quanto em milímetros, as coordenadas cartesianas (x,y) do centróide do último objeto processado. Os

valores exibidos em milímetros são enviados ao programa de manipulação, carregado no controlador do robô.

3.1.3 Algoritmos e Cálculos

Nas rotinas implementadas no módulo de processamento foram utilizados métodos matemáticos selecionados por serem os mais simples e necessitarem de um menor tempo de processamento, conforme descritos no capítulo dois. Abaixo são descritos tais métodos. O Apêndice A ilustra a seqüência de configuração e ativação do SVR e do processo de determinação e envio dos dados dos centróides dos objetos analisados.

Referenciamento do robô

O referenciamento do robô é executado conforme mostra a figura 3.5. Um objeto é colocado na posição $P1$, sendo que seu centróide deve coincidir com o ponto superior esquerdo da janela de visualização de imagens. Uma réplica do objeto, localizada na posição $P1$, é colocada, então, na posição $P2$. Em seguida é realizado o referenciamento do robô, onde se posiciona o efetuator no centróide $P2$. Assim, existe um deslocamento espacial Δy entre as referências da câmera e do robô.

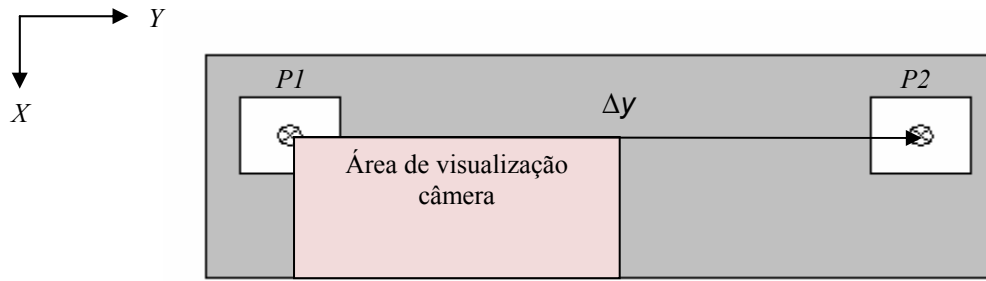


Figura 3.5 – Simulação de movimento da esteira

A posição espacial do centróide de um objeto qualquer é informada ao robô, no formato de coordenadas retangulares (x,y,z) , após seu processamento. Porém, a coordenada final de captura será igual a $(x,y+\Delta y,z)$. Assim, qualquer objeto que teve o seu centróide calculado num ponto qualquer sob a área de visualização da câmera, será capturado pelo robô numa posição onde a coordenada Y está acrescido de Δy , sendo que a coordenada X não se altera.

Simulação do movimento da esteira

Após o referenciamento, ajusta-se o intervalo de tempo que transcorre entre cada execução do processamento da imagem. Além desse tempo, devem ser considerados os tempos de processamento da imagem e de resposta do robô que, somados, resultam em 1,4 segundos, com desvio padrão de 0,3 segundos, conforme o Apêndice B. É importante salientar que o tempo de processamento da imagem depende, de maneira inversamente proporcional, da velocidade do processador utilizado no PC.

Denominando os tempos de processamento e de resposta de t' , e considerando que a velocidade da esteira é conhecida e constante, v , deduz-se a equação para determinação do intervalo de tempo Δt , que deve ser ajustado módulo de processamento, sendo

$$\Delta t = \frac{\Delta y}{v} - t' \quad (11)$$

Determinação do valor da intensidade de cor dos pixels

Após a obtenção da imagem, é iniciado o processo de varredura, píxel por píxel, na janela de visualização do módulo de captura. Essa varredura inicia-se no ponto superior

esquerdo da janela de visualização e termina no ponto superior direito. O processo é reiniciado na linha inferior, e repetido até que todos os pixels da janela de visualização sejam lidos. O objetivo é atribuir a cada píxel um valor numérico proporcional ao valor da intensidade de cor exibida.

Como uma cor qualquer é formada pela combinação das cores da tríade *RGB*, é mais simples atribuir um valor numérico para cada componente da tríade do que para a cor resultante. Como a intensidade da cor de cada um desses componentes é determinada por um número de oito bits, existem apenas 256 tons para cada píxel. Portanto, o sistema identifica a cor e atribui a cada um deles um valor entre zero e 255.

Monocromatização da Imagem

Após a leitura e atribuição de um valor numérico para cada sub-píxel da *tríade*, é necessário transformar a cor resultante em um tom de cinza. Imagens em tons de cinza possuem o mesmo valor de intensidade de cor para os três sub-pixels, pois, apenas assim, a cor resultante do píxel pertencerá à diagonal principal do cubo de cores *RGB*, conforme mostra a figura 2.5

Para que esse valor numérico, que também varia de zero a 255, seja determinado, é utilizada a Equação (3), onde o valor resultante de “*Y*” será o novo valor de intensidade de cor que cada sub-píxel assume. Após realizar essa tarefa para cada píxel, a combinação de cores da imagem resultante será completamente em tons de cinza.

Limiarização da Imagem

Após o processo de monocromatização é necessário limiarizar a imagem, transformando-a em preto-e-branco ou binária. Para isso, é utilizada a Equação (5), onde $f(x,y)$ é o valor de intensidade de cor cada píxel (já monocromatizados) e T é o valor limite determinado. Se o valor da intensidade de cor de qualquer píxel for superior ao limite T , então seu valor é “ligado” ou 1. Se for inferior, é “desligado”, ou zero. Como resultado, obtém-se uma imagem binária ideal para as rotinas de análise.

Cálculo da Área e do centróide

O processo de cálculo da área em pixels, já descrito no capítulo dois, é efetuado utilizando a Equação (6). Adotando-se os valores das constantes a e b como iguais a zero e, por tratar-se de uma imagem binária, a área do objeto é determinada realizando-se apenas um somatório do número total de pixels “ligados” ou 1.

Para o cálculo dos centróides são utilizadas as equações (7) e (8). Os momentos $M_{0,1}$ e $M_{1,0}$ representam os somatórios dos pixels pretos multiplicados por suas distâncias em relação aos eixos imaginários x e y da imagem, respectivamente. Como os valores dos pixels pretos são “ligados” ou 1, basta realizar um somatório das posições dos pixels pretos em relação aos eixos imaginários x e y .

Conversão da unidade do centróide

Os centróides calculados no processo descrito anteriormente possuem a unidade de medida linear “pixels”. É necessário converter essa unidade para milímetros ajustando os dados, que são, então, enviados ao controlador do robô na escala correta. Para que essa conversão seja feita corretamente, é necessário multiplicar os centróides calculados por um fator de conversão calculado através de um processo de calibração que é realizado configurando as posições de um objeto conforme mostra a Figura 3.6.

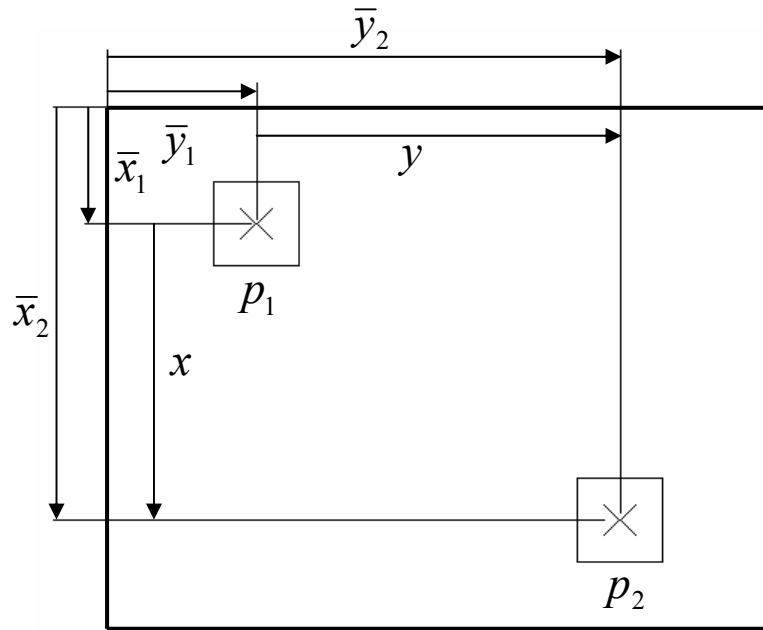


Figura 3.6 – Calibração do fator de conversão de unidades.

Primeiramente, um objeto é colocado na posição p_1 , logo, o processo de determinação do centróide é acionado, e determina os valores \bar{x}_1 e \bar{y}_1 , e, em seguida, desloca-se o objeto para a posição p_2 , deslocando o seu centróide em x e y milímetros. O processo de cálculo do centróide é acionado novamente, e são obtidos os valores \bar{x}_2 e \bar{y}_2 . Os valores dos fatores de transformação para o eixo X e para o eixo Y , respectivamente x_f e y_f , são calculados por:

$$x_f = \frac{x}{\bar{x}_2 - \bar{x}_1} \quad (9)$$

$$y_f = \frac{y}{\bar{y}_2 - \bar{y}_1} \quad (10)$$

Os valores calculados de x_f e y_f são, então, inseridos nos campos “1 *pixel* = x mm” e “1 *pixel* = y mm”, respectivamente.

Envio dos dados para o controlador do robô

Os valores das coordenadas do centróide da imagem processada são enviados ao programa de manipulação carregado no controlador do robô, através do módulo de comunicação IRBCom. As rotinas de envio de dados utilizadas, indicadas por Bayer, 2004, são:

```
IRB_SendCommand (10)
IRB_SendReal ()
```

A primeira rotina, que prepara o robô para realizar a movimentação em direção ao centróide, envia o comando “*move*”, indicando ao programa de manipulação que mais três variáveis numéricas serão enviadas.

Essas variáveis são enviadas pela segunda rotina repetida três vezes, indicando, respectivamente, as coordenadas (x,y,z) . Esses números informam o quanto o efetuador do robô terá de se mover em relação à posição de referência para que possa capturar o objeto.

Calibração da Área

Uma propriedade importante do módulo de processamento é discernir se, sob o campo de visão, há um objeto completa ou parcialmente visualizado, ou se ainda há mais de um objeto. Isso é realizado utilizando uma comparação do valor da área do objeto processado com o valor de área obtido pelo processo de calibração.

Para tanto, coloca-se o objeto sob a câmera e pressiona-se o botão “*calibrar área*”. O programa repete a rotina de cálculo da área dez vezes e calcula a área média. Os limites máximos e mínimos dessa área são calculados utilizando-se três desvios-padrões, para mais ou para menos, obtendo assim uma confiabilidade de 99,87%, segundo a curva normal de distribuição de probabilidades. O desvio padrão percentual estimado para o processo é 0,26%, conforme Apêndice C, (multiplicado por três obtêm-se um limite de 0,78%). Porém, por questões de praticidade, utiliza-se no programa o limite de 1,0%. É importante salientar que a precisão do sistema é sensível a variação das condições de iluminação do ambiente.

Fluxograma dos processos

A Figura 3.7 demonstra a seqüência de configuração e ativação do SVR e do processo de determinação e envio dos dados dos centróides dos objetos analisados.

3.2 Programa de Manipulação

O programa de manipulação de objetos, carregado no controlador do robô, consiste numa seqüência de comandos e que reconhece as informações enviadas pelo módulo de processamento. Sendo o robô empregado no desenvolvimento deste trabalho um modelo ABB IRB1400, o programa de manipulação foi desenvolvido em linguagem ABB Rapid. Para reconhecimento de dados, foram adicionadas as rotinas descritas por Bayer, 2006, específicas em seu código fonte.

Diferentemente de um programa comum de manipulação comum, dados como a posição do objeto a ser capturado não estão incluídos no código fonte do programa. Ao invés disso, o programa apresenta variáveis, que são os dados enviados pelo PC.

O programa SVR.prg executado no controlador do robô tem como elemento central um laço que faz a leitura do comando, identifica-o e chama a rotina solicitada. Essa rotina fica responsável por receber os dados e executar a tarefa para a qual foi programada.

Concluída a tarefa, o programa retorna ao laço central e chama *ReceiveByte* para receber o próximo comando e, caso o PC não tenha nenhum comando a ser enviado, o controlador permanece aguardando. Do mesmo modo, no intervalo de tempo em que a tarefa está sendo executada, o controlador não é capaz de receber outros comandos, pois o PC verifica que o controlador está indisponível e mantém o sistema aguardando.

3.3 Dispositivos de Captura de Imagens

O dispositivo de captura de imagens é formado pelo uso conjunto da câmera e do suporte. A câmera escolhida foi do tipo digital, que são as mais utilizadas atualmente e são facilmente encontradas no mercado. Além disso, permite uma rápida interface com os demais sistemas que são empregados no projeto, gerando um sinal digital facilmente captado por aplicativos de visualização.

No presente trabalho, é utilizado a *webcam Microsoft VX1000*, colorida, conforme mostra a Figura 3.7. Essa câmera foi selecionada porque possui simples operação, dispensa uma placa de captura de imagens devido à sua entrada USB e possui resolução máxima de 640x480 pixels e taxa de atualização máxima de 30 quadros por segundo.



Figura 3.7 – Webcam Microsoft VX100 e seu cabo com entrada USB.

O suporte da câmera, conforme a Figura 3.8, estava disponível no laboratório de robótica, sendo necessário apenas desenvolver um adaptador para acoplá-la. Esse adaptador foi fabricado nos equipamentos do laboratório utilizando nylon.

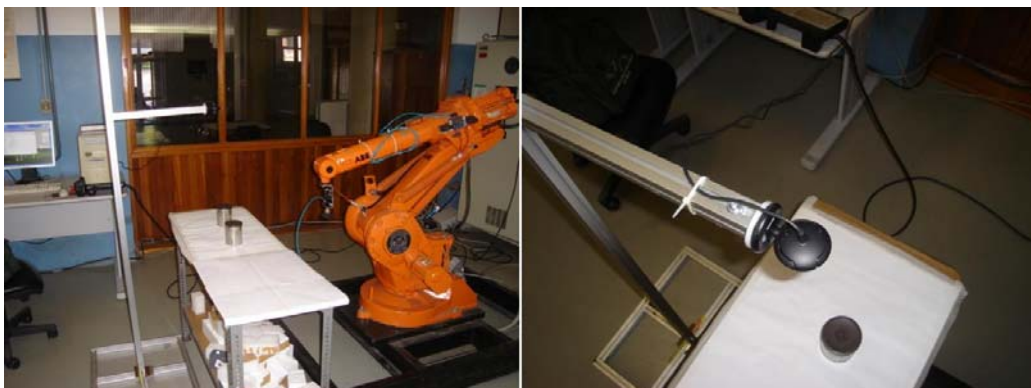


Figura 3.8 – Vista lateral e superior da *webcam* acoplada ao suporte.

3.4 Iluminação

Os tipos de iluminação normalmente empregados em sistemas de visão robótica destinados à manipulação são diretas ou traseiras. Após a captação de algumas imagens sobre a área de trabalho, com o intuito de definir a melhor maneira de iluminá-la, constatou-se que as luminárias do laboratório fornecem iluminação suficiente sobre a área de trabalho sem gerar sombras ou qualquer tipo de interferência. Sendo assim, não há a necessidade de se realizar qualquer alteração na iluminação.

3.5 Módulo IRBCom

Devido à incompatibilidade dos sinais elétricos entre as duas interfaces utilizadas, é necessário utilizar um dispositivo que ajuste os níveis de tensão e garanta a segurança de funcionamento na comunicação entre os equipamentos. O dispositivo IRBCom, mostrado na Figura 3.10, é formado por uma série de acopladores óticos de alta sensibilidade, que permitem a transmissão dos dados sem nenhuma conexão elétrica entre a porta paralela do PC e a Placa E/S do controlador do robô. Os circuitos foram montados em placas de circuito impresso e acondicionados em um gabinete com conectores externos. O painel frontal do gabinete possui leds indicadores de sinal em cada linha de comunicação.



Figura 3.10 – Módulo IRBCom.

4. AVALIAÇÃO EXPERIMENTAL

Depois de concluídos o desenvolvimento e a configuração do sistema, iniciou-se a avaliação experimental para averiguar sua praticidade, eficiência, e conseqüente aplicabilidade. Os equipamentos utilizados para o experimento foram um computador pessoal padrão IBM-PC, o módulo de comunicação IRBCom, o robô ABB IRB1400 e o gerador de vácuo Schrader Bellows (ventosa), que possui acionamento pneumático controlado por um sinal elétrico gerado pelo robô, gerando vácuo quando o sinal está ativado.

Como corpos de prova para a avaliação do programa, foram empregados dois sólidos de isopor, no formato de blocos de seção retangular. Como já descrito anteriormente, para o processamento de imagens, foi utilizado um sólido com a superfície escura sobre uma superfície clara.

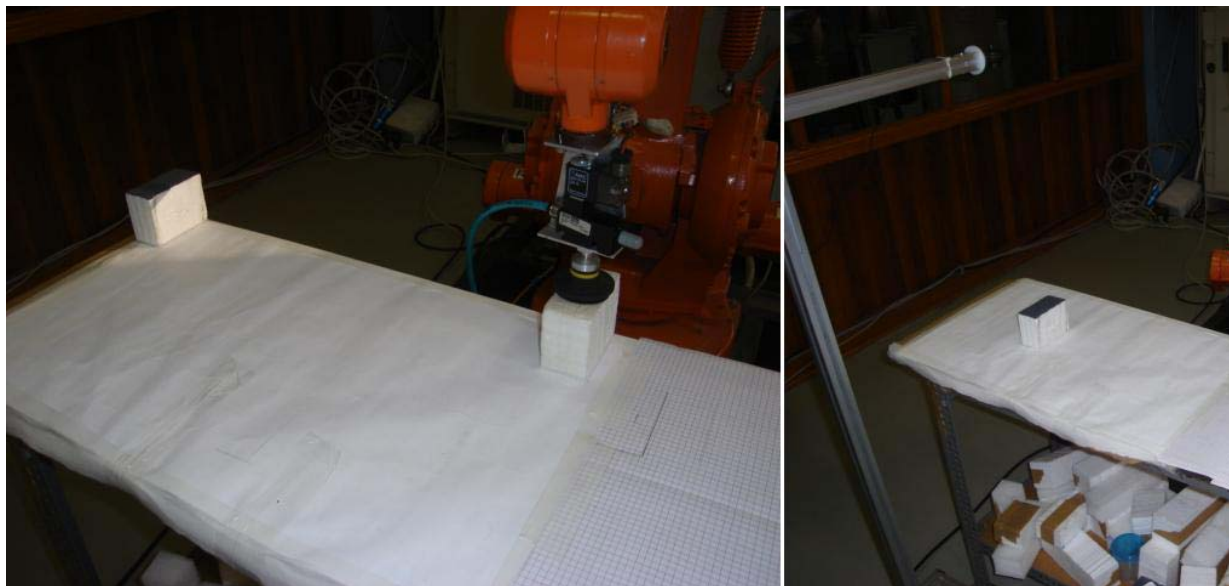


Figura 4.1 – Processos de referenciamento e calibração da área.

O experimento iniciou-se pelo referenciamento do robô e calibração da área, conforme mostra a Figura 4.1. Conforme descrito anteriormente, esses são processos críticos para o funcionamento coerente do sistema, pois um pequeno erro pode resultar no mau posicionamento do robô ou da avaliação da área da peça. O deslocamento utilizado entre os dois centróides das peças de referenciamento foi de seiscentos milímetros na direção do eixo Y do robô.

Após a realização desses processos, é necessário configurar o posicionamento das peças com a finalidade de simular o efeito de movimento da esteira. Para isso, posicionou-se uma peça em uma posição aleatória da base, mas sob o campo de visão da câmera. Uma segunda peça, com as dimensões idênticas a da primeira, é colocada em uma posição com o mesmo deslocamento em relação ao eixo y do robô, utilizado no processo de referenciamento. Seqüencialmente, é iniciado o processo de captura e processamento das imagens.

5. RESULTADOS

Os resultados obtidos no processo experimental foram satisfatórios, atingindo os objetivos propostos no presente trabalho. O sistema obteve êxito na tarefa de identificar o centróide de peças de diversas geometrias e enviar as informações ao controlador do robô, sendo que esse, por sua vez, executou as tarefas de manipulação, conforme mostra a Figura 5.1.



Figura 5.1 – Captura e manipulação da peça.

Os procedimentos de captura, processamento e análise da imagem são feitos de forma ágil pelo programa SVR, assim como a geração, transmissão e recepção de sinal. O programa de manipulação SVR.prg, carregado no controlador do robô, executa a recepção dos comandos de forma rápida permitindo a eficiente e precisa operação do manipulador.

O processo de calibração da área mostrou-se bastante eficiente e com boa confiabilidade. Porém, qualquer variação na condição de iluminação do ambiente pode ocasionar um erro no processamento da imagem.

As falhas do sistema, que foram evidenciadas durante a avaliação, consistem em alguns erros de posicionamento do efetuador para a captura do objeto. Esses erros são detectados quando o referenciamento é feito de forma imprecisa pelo operador, somadas às imprecisões de posicionamento das peças sobre a mesa.

6. CONCLUSÃO

Os resultados obtidos atendem aos objetivos, comprovando que sistemas de visão mais simples, desenvolvidos em computadores pessoais, podem gerar resultados com boa precisão de localização. Além disso, mostrou-se ser de fácil implementação, visto que necessita apenas uma *webcam* (amplamente disponível no mercado) conectada a um PC, compatível com o ambiente *Windows*, que executa os módulos de captura e processamento de imagens.

Para a transmissão de sinais em tempo real utiliza-se um dispositivo prático e de fácil programação, também desenvolvido no Laboratório de Robótica, que utiliza a porta paralela disponível em PC e placa E/S existente no módulo de controle do robô. O programa de manipulação demonstrou igual desempenho satisfatório, mas possui funcionamento restrito apenas para robôs ABB, visto que utiliza linguagem Rapid nativa desses manipuladores. Para outros manipuladores, o sistema funciona igualmente com um programa equivalente no controle.

Portanto, o presente sistema identifica-se como uma alternativa viável para o controle do robô através de um microcomputador PC, apresentando alta confiabilidade e baixo custo de implementação em operações de manipulação. Em contra partida, salienta-se a importância do desenvolvimento de um sistema de visualização gráfica da imagem processada e de seu respectivo centróide, ou, ainda, o desenvolvimento de um programa semelhante, mas que realize o processamento da imagem no domínio-frequência, sendo essas possíveis sugestões para futuros trabalhos.

REFERÊNCIAS BIBLIOGRÁFICAS

Bayer, Fernando Mariano. **“Desenvolvimento e Implementação de uma Interface de Comunicação para Controlador Robótico Industrial”**, 2004. Dissertação (Programa de Pós Graduação em Engenharia Mecânica) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2004.

Fu, K. S., Gonzalez, R. C., Lee, C. S. G., 1987. **“Robotics: Control, Sensing, Vision and Intelligence”**, McGraw-Hill, New York.

Gonzalez, R. C., Woods, R. E., 2003. **“Processamento de Imagens Digitais”**, Editora Edgard Blücher, São Paulo.

Gurewicz, O., Gurewicz, N., 1997. **“Aprenda em 21 Dias Visual Basic 4”**, Editora Campus, Rio de Janeiro.

Grassi, Mauricio Veloso. **“Desenvolvimento e Aplicação de um Sistema de Visão para o Robô Industrial de Manipulação”**, 2005, Dissertação (Programa de Pós Graduação em Engenharia Mecânica) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2005.

Okigawa, M., 1992. **“A New and Simple Test Structure for Evaluating the Sectional Photo Sensitivity Distribution of Pixels in a Frame-Transfer CCD Image Sensor”**, Proceedings of IEEE 1992 International Conference of Microelectronic Test Structures, vol. 5, pp. 134-138.

Perry, G. M., 1999. **“Aprenda em 21 Dias Visual Basic 4”**, Editora Campis,

Simões, A. S., Costa, A. H. R. **“Redes Neurais Pulsadas de Base Radial com Treinamento não Supervisionado: Um Estudo da Capacidade de Agrupamento para a Visão Robótica”**, VII SBAE / II IEEE LARS, São Luiz, 2005.

Torres, G., 2001. **“Hardware – Curso Completo”**, Axcel Books.

BARA, www.bara.org.uk/encyclopedia/robots/robintro.doc. Acessado em 17/10/2007.

IFTOMM, www.caip.rutgers.edu/iftomm. Acessado em 17/10/2007.

InfoWester, <http://www.infowester.com/monitores.php>. Acessado em 27/09/06.

Macoretti, <http://www.macoratti.net/webcam.html>. Acessado em 22/09/2007.

RIA, 17/10/2007 www.roboticsonline.com

Técnicas de Análise de Imagens para Robótica, <http://www.inf.ufsc.br/~visao/robotica.html>. Acessado em 08/08/2007.

APÊNDICE A

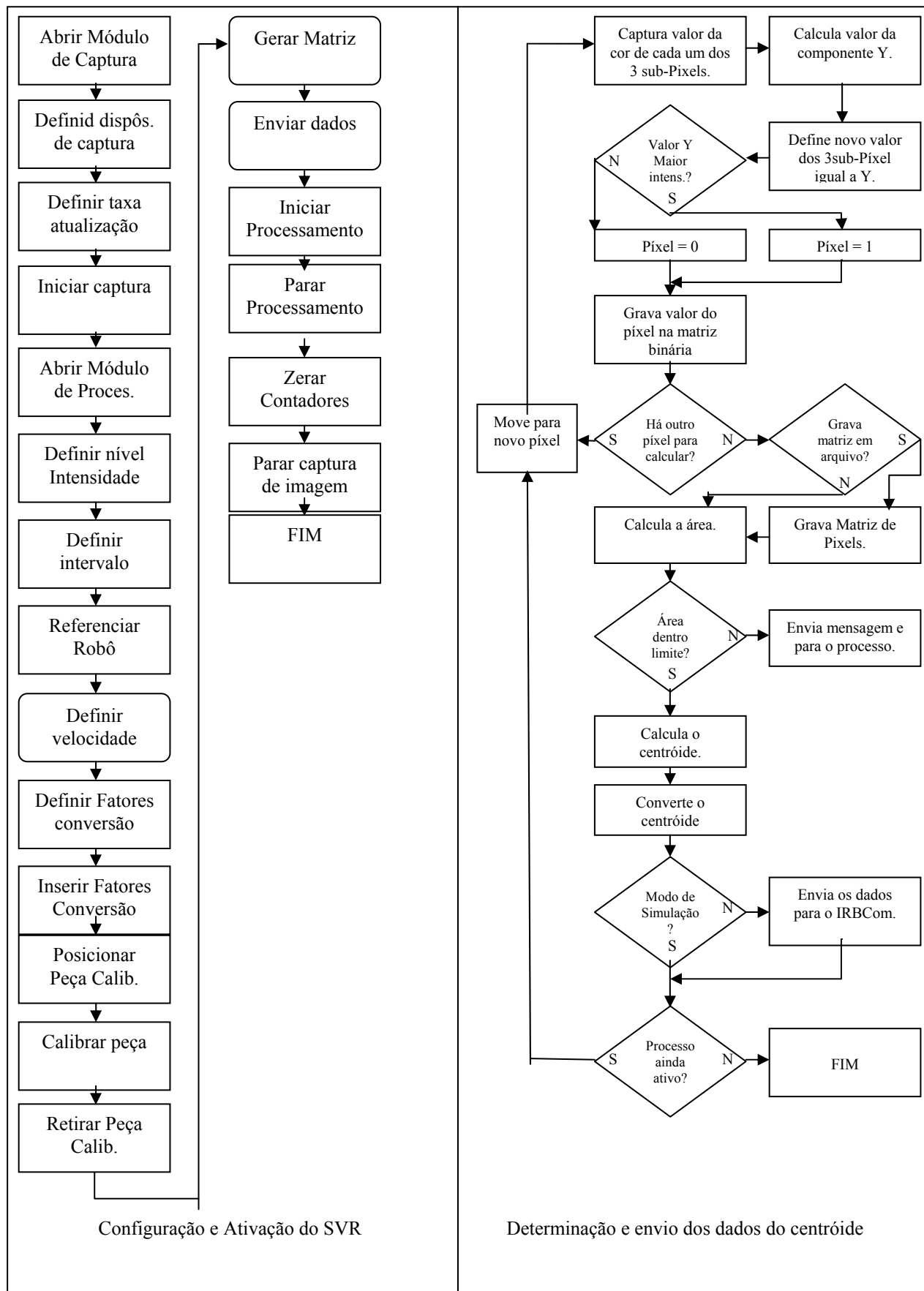


Figura 1 – Configuração SVR e processo de cálculo de centróides.

APÊNDICE B

Estimativa do tempo de resposta do robô.

O tempo médio de resposta do sistema foi estimado em 1,37 segundos, com desvio padrão de 0,29 segundos. A Tabela 1 exibe os resultados das calibrações realizadas. Foram selecionados sete intervalos diferentes de tempo entre a execução de cada imagem: 1, 3, 5, 7, 10, 15 e 20 segundos. O Tempo Total de Processamento (TTP) é a soma do intervalo de tempo selecionado e do Tempo de Resposta do Sistema (TRS). Nota-se que o TRS é similar para qualquer intervalo de tempo selecionado.

A tabela abaixo foi obtida realizando-se 25 cálculos sequenciais da área de uma determinada peça, para os intervalos de tempo citados.

Tabela 01 – Tempos de Resposta do Sistema.

Número de Processos	Tempo Selecionado													
	1		3		5		7		10		15		20	
	TTP	TRS	TTP	TRS	TTP	TRS	TTP	TRS	TTP	TRS	TTP	TRS	TTP	TRS
1	2,4	1,4	4,8	1,8	6,4	1,4	8,3	1,3	11,6	1,6	16,7	1,7	21,6	1,6
2	2,8	1,8	4,2	1,2	6,8	1,8	8,7	1,7	11,0	1,0	16,1	1,1	20,9	0,9
3	2,2	1,2	4,6	1,6	6,2	1,2	8,1	1,1	11,4	1,4	16,4	1,4	21,4	1,4
4	2,6	1,6	3,9	0,9	6,5	1,5	8,6	1,6	11,8	1,8	16,9	1,9	21,8	1,8
5	2,1	1,1	4,4	1,4	6,0	1,0	8,0	1,0	11,2	1,2	16,2	1,2	21,2	1,2
6	2,6	1,6	4,8	1,8	6,4	1,4	8,4	1,4	11,6	1,6	16,6	1,6	21,6	1,6
7	2,0	1,0	4,2	1,2	6,8	1,8	8,7	1,7	11,0	1,0	16,1	1,1	20,9	0,9
8	2,5	1,5	4,6	1,6	6,1	1,1	8,2	1,2	11,4	1,4	16,4	1,4	21,3	1,3
9	1,9	0,9	4,0	1,0	6,6	1,6	8,5	1,5	11,8	1,8	16,8	1,8	21,8	1,8
10	2,3	1,3	4,4	1,4	6,0	1,0	7,9	0,9	11,2	1,2	16,3	1,3	21,2	1,2
11	2,7	1,7	4,8	1,8	6,4	1,4	8,3	1,3	11,6	1,6	16,6	1,6	21,6	1,6
12	2,0	1,0	4,2	1,2	6,8	1,8	8,7	1,7	11,0	1,0	16,0	1,0	21,0	1,0
13	2,4	1,4	4,7	1,7	6,2	1,2	8,0	1,0	11,4	1,4	16,4	1,4	21,3	1,3
14	2,7	1,7	4,0	1,0	6,6	1,6	8,5	1,5	11,8	1,8	16,8	1,8	21,7	1,7
15	2,1	1,1	4,4	1,4	6,0	1,0	7,8	0,8	11,2	1,2	16,2	1,2	21,3	1,3
16	2,5	1,5	4,8	1,8	6,4	1,4	8,2	1,2	11,5	1,5	16,6	1,6	21,7	1,7
17	2,0	1,0	4,1	1,1	6,7	1,7	8,6	1,6	11,3	1,3	16,0	1,0	21,0	1,0
18	2,5	1,5	4,6	1,6	6,2	1,2	8,1	1,1	11,8	1,8	16,3	1,3	21,5	1,5
19	2,0	1,0	4,1	1,1	6,6	1,6	8,5	1,5	11,2	1,2	16,7	1,7	20,9	0,9
20	2,4	1,4	4,5	1,5	5,9	0,9	7,9	0,9	11,6	1,6	16,2	1,2	21,3	1,3
21	2,9	1,9	3,9	0,9	6,3	1,3	8,3	1,3	11,0	1,0	16,5	1,5	21,6	1,6
22	2,3	1,3	4,2	1,2	6,7	1,7	8,7	1,7	11,4	1,4	15,9	0,9	21,0	1,0
23	2,7	1,7	4,6	1,6	6,1	1,1	8,0	1,0	11,8	1,8	16,3	1,3	21,4	1,4
24	2,1	1,1	4,0	1,0	6,5	1,5	8,5	1,5	11,2	1,2	16,7	1,7	21,9	1,9
25	2,4	1,4	4,5	1,5	5,8	0,8	8,8	1,8	11,7	1,7	16,1	1,1	21,2	1,2
Média	1,36		1,37		1,36		1,33		1,42		1,39		1,36	
Desvio	0,29		0,30		0,30		0,30		0,28		0,29		0,31	

APÊNDICE C

Estimativa dos limites da área de calibração

Para determinar os limites percentuais que o programa utiliza, foram realizados e anotados 45 procedimentos de cálculo de área, em pixels, de corpos de prova diferentes. Os resultados são mostrados na Tabela 2.

Tabela 2 – Resultados obtidos para as áreas.

Área Média (em Pixels)			
Cálculo	Peça 1	Peça 2	Peça 3
1	23925	14924	14809
2	24190	14921	14812
3	24090	14900	14806
4	24067	14897	14802
5	24095	14913	14801
6	24080	14932	14801
7	24152	14914	14799
8	24137	14901	14794
9	24125	14921	14798
10	24166	14928	14823
11	24097	14917	14805
12	24181	14925	14797
13	24113	14913	14802
14	24216	14921	14800
15	24182	14902	14806
16	24200	14906	14803
17	24127	14915	14800
18	24237	14899	14806
19	24172	14914	14810
20	24133	14932	14806
21	24102	14918	14796
22	24163	14875	14809
23	24084	14921	14792
24	24072	14911	14801
25	24019	14907	14805
26	24137	14906	14788
27	24115	14868	14793
28	24076	14887	14794
29	24109	14865	14805
30	24180	14899	14801
31	24166	14928	14798
32	24136	14912	14792
33	24112	14894	14794
34	24101	14937	14798
35	24115	14908	14795
36	24084	14900	14796
37	24000	14890	14803
38	24056	14905	14802
39	24088	14896	14811
40	24090	14896	14804
41	23995	14894	14808
42	24035	14889	14799
43	24064	14892	14793
44	24046	14875	14796
45	24036	14900	14808

Em seguida, foram calculadas a área média, o desvio padrão, a variação máxima e o desvio padrão percentual, para cada corpo de prova, conforme ilustra a tabela 3. Os resultados obtidos no processamento da “Peça 1” foram utilizados para estimar o desvio-padrão percentual que será utilizado pelo programa de processamento de imagens.

Tabela 3 – Dados calculados dos 03 corpos de prova.

Dados Calculados	Peça 1	Peça 2	Peça 3
Área Média	24108	14906	14801
Desvio Padrão	62	17	7
Maior Área	24237	14937	14823
Menor Área	23925	14865	14788
Variação Máxima	312	72	35
Desvio Padrão (%)	0,259%	0,113%	0,045%

Agradecimentos

Em primeiro lugar agradeço à Cláudia, minha eterna fonte de inspiração.

à minha família. Sem eles, nada disso seria possível.

ao meu orientador Prof. Flavio José Lorini, pelo tempo e atenção dedicados a esse trabalho.

à universidade e aos professores do Departamento de Engenharia Mecânica.

Lemos, N. C. **Desenvolvimento de um Sistema de Visão para Robô Industrial aplicado na manipulação de objetos reconhecidos em esteiras transportadoras em movimento. 2007.** 31f. Monografia (Trabalho de Conclusão do Curso de Engenharia Mecânica) – Departamento de Engenharia Mecânica, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2007.

RESUMO

O presente trabalho propõe o desenvolvimento de um sistema de visão (aquisição e tratamento de imagem) para aplicação associada a um robô industrial antropomorfo denominado SVR – Sistema de Visão Robótica. O objetivo é reconhecer diferentes objetos em posições aleatórias dentro do campo de cobertura de uma câmera. Definidas as coordenadas de posição, inicia-se o processo de transmissão dessas ao controle do robô para gerar a trajetória necessária à manipulação dos objetos reconhecidos. O aplicativo foi desenvolvido em linguagem *Visual Basic* utilizando rotinas de transmissão de dados já disponíveis no Laboratório de Robótica do DEMEC-UFRGS. São utilizados equipamentos de captura de imagem integrados a um computador interligado ao controle do robô. Cabe ressaltar ainda que o sistema de visão desenvolvido caracteriza-se pela sua facilidade de implementação e aplicação em tarefas de manipulação robótica industrial. O programa é utilizado em situações que exijam a determinação da posição de objetos em um ponto de uma esteira em movimento, no campo do sistema de visão, dentro de uma porção do volume de trabalho do robô. Os resultados obtidos atendem aos objetivos propostos, comprovando que sistemas de visão mais simples, desenvolvidos em computadores pessoais, podem gerar resultados com boa precisão de localização. O presente sistema identifica-se como uma alternativa viável para o controle do robô através de um microcomputador (padrão PC), apresentando alta confiabilidade e baixo custo de implementação em operações de manipulação.

PALAVRAS-CHAVE: Robô, visão robótica, robótica industrial.

Lemos, N. C. **Vision system development for industrial robot applied to the manipulation of recognized objects on moving conveyors.** 2007. 31f. Monografia (Trabalho de Conclusão do Curso de Engenharia Mecânica) – Departamento de Engenharia Mecânica, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2007.

ABSTRACT

The present monograph sets out to develop a vision system (acquisition and treatment of image) to be practice connected to an anthropomorphic industrial robot named RVS — Robotic Vision System. The objective is to recognize different objects randomly positioned inside the camera coverage. After having been delimited, the coordinates are transmitted to the robot's control in order to define the route to the manipulation of the recognized objects. The application was developed in *Visual Basic* language along with the transmission of data routines already in use in DEMEC-UFRGS's Robotics Laboratory. In addition, image capturing tools integrated to a computer connected to the robot's control were used. This vision system is characterized by its facility of implementation and application in industrial robotics manipulation tasks. The application is used in situations that demand the determination of object position in a certain point of a moving conveyor, within the vision system field, and within a portion of the volume of the robot's work. The results obtained support the monograph's objectives, confirming the hypothesis that simpler vision systems, developed in personal computers, can produce results with great location accuracy. This system has proved to be a viable alternative for controlling the robot through a personal microcomputer (PC standard), presenting high reliability and implementation at a low cost in manipulation processes.

KEYWORDS: Robot, robotic vision, industrial robotics.