CONTROLE DE UM MANIPULADOR ROBÓTICO EM UMA TAREFA DE *PICK AND PLACE* AUXILIADO POR UM SISTEMA DE VISÃO

Álvaro Manoel de Souza Soares
Universidade de Taubaté – Departamento de Engenharia Mecânica, Rua Daniel Danelli, s/n – Jardim Morumbi, CEP: 12060-440 – Taubaté - SP. Brasil.
e-mail: alvaro@unitau.br

Valdeci Donizete Gonçalves
Universidade de Taubaté – Departamento de Engenharia Mecânica, Rua Daniel Danelli, s/n – Jardim Morumbi, CEP: 12060-440 – Taubaté - SP. Brasil.
e-mail: valdeci@horizon.com.br

Resumo. *Este trabalho mostra um manipulador robótico executando uma tarefa de "pick and place" auxiliado por um sistema de visão. Um aparato experimental foi montado, composto por um manipulador robótico didático "Lynxmotion" de quatro graus de liberdade e uma "web cam". O espaço de trabalho do robô foi mapeado usando uma "look up table" (LUT), para minimizar os erros de posição do manipulador. A idéia principal é que o robô deve estar apto a apanhar, por exemplo, um parafuso, em qualquer ponto de seu espaço de trabalho com o auxílio do sistema de visão. No espaço de trabalho há um recipiente que fica em uma posição conhecida e um parafuso que pode ser colocado em qualquer lugar. Inicialmente, o sistema de visão adquire a cena da imagem do espaço de trabalho e localiza o centro de gravidade do parafuso. Com esta informação, o robô apanha o parafuso e o coloca no recipiente. Para diminuir o custo experimental do sistema usou-se, uma "web cam" conectada a uma porta "USB" (Universal Serial Bus) do computador. Um programa de computador, em linguagem* C^{++} *foi desenvolvido para adquirir e tratar as imagens da cena e controlar o robô. O programa implementa os algoritmos de processamento de imagem e a cinemática direta do robô. A precisão e repetibilidade são discutidas e os resultados mostram que os algoritmos de controle do robô e processamento de imagens trabalham muito bem.*

Palavras-chave: *Robótica, Sistema de visão e Processamento de Imagem.*

1. INTRODUÇÃO

O processamento de imagens é um campo que cresceu rápido nos últimos anos. As imagens são usadas em diversas áreas como geografia onde a maioria das imagens vem de satélites e aviões e estuda problemas urbanos relacionados à ocupação humana, em robótica com reconhecimento de padrões, na arqueologia, tratando brilho e contraste de imagens para a melhoria de imagens antigas. Na literatura, encontram-se trabalhos como o de Kabayama, A M. e Trabasso L. G. (2002), que descreve três técnicas diferentes de visão computacional no cálculo de medidas em três dimensões onde o *software* MATLAB foi usado para implementar os algoritmos de visão computacional.

Motta e McMaster (2002), apresentam um sistema de medida para fazer a calibração local de um sistema de visão e melhorar a precisão no posicionamento de robôs, Rillo, A. H. R. C. (1996) discute a visão computacional, usando um sistema que inclui percepção visual e um manipulador robótico. Stemmer, M. R. e Neto, C. A. M. (1998) mostram um sistema de reconhecimento de imagem que usa descritores de Fourier e redes neurais para classificar os objetos. Brown (1998) mostra o controle de uma poça de soldagem, onde a geometria da peça de soldagem é medida usando-se um sistema de visão.

Este trabalho utiliza uma configuração experimental composta por um *web cam*, com 320×240 de resolução espacial e 256 níveis cinzas e um *kit* manipulador de robô didático "Lynxmotion 5 Robotic Arm" (Figura 1). Seu propósito é executar uma tarefa de *pick and place* onde o manipulador robótico é auxiliado por um sistema de visão. A cena da câmera é o espaço de trabalho do robô. Dessa forma, o manipulador pode "ver" seu espaço de trabalho usando técnicas de Processamento de Imagem. A cena é adquirida repetidamente e uma subtração entre duas imagens consecutivas é feita a fim de se descobrir se houve movimento na cena. Qualquer objeto colocado na cena, passa a ser a única imagem, resultado da subtração de imagens. Assim o limiar da imagem resultante é calculado e uma imagem binárizada é obtida. O centróide da imagem binarizada é calculado, obtendo-se assim o ponto aonde se encontra o objeto na cena. A cinemática direta do manipulador é utilizada para transformar este ponto para o espaço das juntas do manipulador. Uma vez feita esta transformação, o manipulador robótico, montado em uma configuração de três graus de liberdade, apanha a peça e a coloca em um recipiente. Algoritmos para cálculo do limiar e cinemática direta do robô são discutidos. A correta iluminação da cena, e problemas causados pela iluminação também são discutidos. O programa, desenvolvido em linguagem C++ , trabalha muito bem e ilustra como as áreas (Robótica e Processamento de Imagens) podem trabalhar juntas.

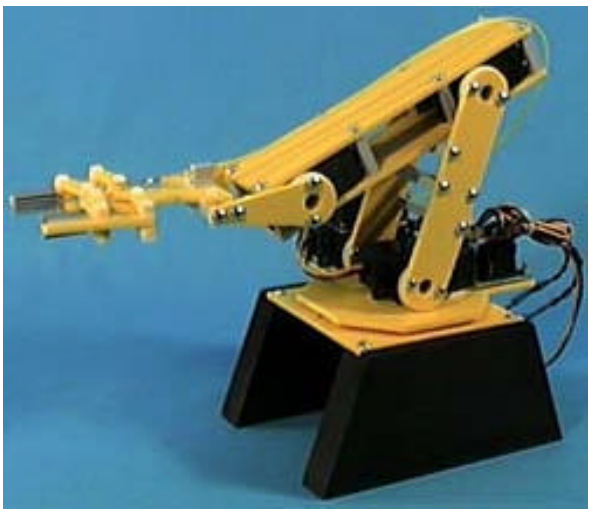

Figura 1 - Manipulador robótico didático Lynx 5

2. O APARATO EXPERIMENTAL

A configuração experimental (Figura2) é composta por uma *web cam* com resolução espacial de 320×240 *pixels* e 256 níveis cinzas e um kit robótico didático *Lynx 5 Robotic Arm*. Um programa de computador, em linguagem Visual C++ da Microsoft foi desenvolvido para controlar o robô e para adquirir e tratar as imagens. A *web cam* está ligada ao PC usando uma conexão USB (Universal Serial Bus) e o manipulador robótico é conectado ao PC através de uma porta serial RS-232.

Figura 2 - Protótipo desenvolvido para o sistema de visão robótica

3. O KIT DIDÁTICO *LINX 5 ROBOTIC ARM*

O *kit* didático *Linx 5* é comercializado pela *Lynxmotion Inc*., possui 4 graus de liberdade, compostos por quatro juntas revolutas mais a garra. Seus atuadores são servo-motores. Neste trabalho, com o intuito de simplificação, foram usados três graus de liberdade revolutos mais o movimento da garra. O quarto ângulo foi fixado constante e igual a 40 graus. Desenvolveu-se a cinemática direta do manipulador, com o objetivo de se transformar as coordenadas do espaço de trabalho do manipulador em coordenadas nos espaços das juntas.

A configuração do manipulador é simples e os parâmetros de Denavit-Hartenberg para esta configuração são descritos na tabela 1.

Tabela 1. Parâmetros de Denavit-Hartenberg

Elo	θ_i	α_i	a_i	d_i
1	θ_1	0	0	d_1
2	θ_2	$p/2$	0	0
3	θ_3	0	a_3	0
4	θ_3	0	a_4	0

Uma vez descritos tais parâmetros, as expressões que compõe a cinemática direta do manipulador foram obtidas através da multiplicação das matrizes de transformação homogêneas e são descritas a seguir.

$$x = 5.08\cos(-q_4 + q_1 + q_2 - q_3) + 5.08\cos(q_4 + q_1 + q_2 - q_3) + 5.08\cos(q_1 + q_2 - q_3) + 5.08\cos(q_1 + q_2 - q_3) \quad (1)$$

$$y = 5.08\,sin(q_4 + q_1 + q_2 + q_3) + 5.08\,sin(-q_4 + q_1 + q_2 - q_3) + 5.08\,sin(q_1 + q_2 + q_3) + 5.08\,sin(q_1 + q_2 - q_3) \quad (2)$$

$$z = 10.16\,sin(q_3 + q_4) + 7.62 + 10.16\,sin(q_3) \quad (3)$$

Através da cinemática direta do manipulador, gerou-se uma *Look Up Table* (LUT) de forma que os correspondentes valores dos ângulos θ_1, θ_2 e θ_3 fossem obtidos, a partir das coordenadas cartesianas, sem a necessidade do cálculo da cinemática inversa do manipulador.

4. O PROGRAMA DESENVOLVIDO

O programa desenvolvido, ao ser iniciado, adquiri imagens da cena, afim de separar o fundo da cena dos objetos que a compõe. Seus passos são ilustrados no fluxograma da Figura 3. Em um

passo seguinte, ocorre o processamento das imagens adquiridas. Esta parte é responsável pelo processamento da imagem e por obter a posição do objeto remanescente na cena em coordenadas cartesianas e por disponibilizar esta informação para o módulo do programa que trata do controle do manipulador. A rotina do programa de controle do manipulador é responsável pela coleta das coordenadas cartesianas e pela movimentação do manipulador fisicamente para esta posição. Também fica contida nesta parte a rotina de programação do robô, para que ele pegue o objeto (fechar a garra), e retornar à posição inicial, onde ficará aguardando um próximo evento.

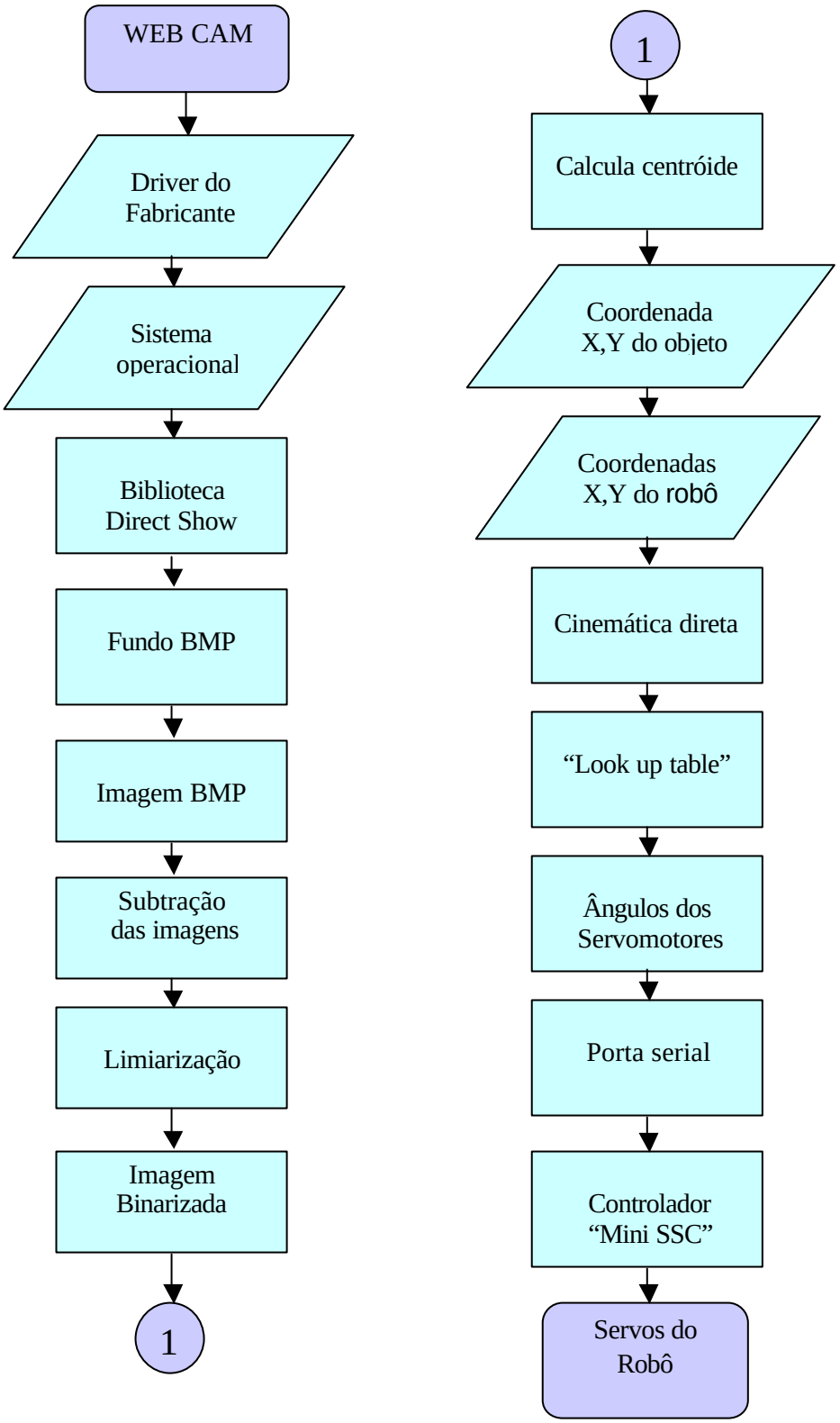

Figura 3 - Fluxograma do programa desenvolvido

4.1. O Sistema de Visão

O sistema de visão utilizado neste trabalho é composto por uma *web cam*, responsável pela captura das imagens. Este tipo de câmera possui baixo custo e prescindi de uma placa de captura de imagens (*frame grabber*), simplificando ainda mais seu uso. Sua comunicação com o computador é feita usando-se uma porta USB e a qualidade das imagens adquiridas, foi satisfatória para o trabalho em questão. Para o interfaceamento entre a *"web cam"* e o computador, foram usadas rotinas da biblioteca *"Direct Show"* da Microsoft©.

O programa tem como entrada as imagens adquiridas pela *web cam*. Inicialmente, são armazenadas duas imagens da cena (imagem.bmp e fundo.bmp), que serão utilizadas posteriormente. As imagens seguintes são adquiridas e binarizadas através do cálculo do limiar da imagem, para que se possa calcular a posição cartesiana do centróide do objeto remanescente na cena.

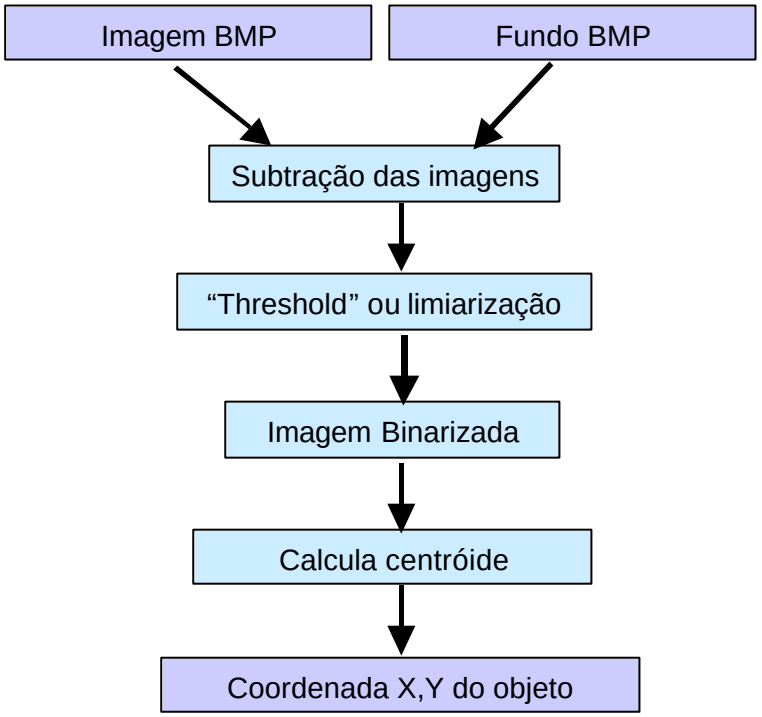

Figura 4 – Sistema de visão

Na subtração das imagens são considerados os valores "pixel" a "pixel", tendo-se como resultado a imagem do objeto, que é colocado na área de trabalho do sistema de visão já com 256 tons de cinza. Em seguida é feito o cálculo do limiar da imagem e sua posterior binarização, onde cada pixel da mesma terá somente valores 0 (zero) ou 255. Após cálculo do limiar, é feito o cálculo de sua área e da centróide em coordenadas cartesianas. Os valores das coordenadas cartesianas são então enviados para a rotina do programa responsável por movimentar o manipulador robótico até a posição desejada.

4.2. Movimentação do Manipulador Robótico

O processo do recebimento das coordenadas X,Y da imagem e transformadas nas coordenadas X,Y do robô, ou seja, para onde o robô irá se deslocar para pegar o objeto e capturá-lo, levando-o para a posição desejada. As coordenadas dos pontos do robô foram obtidas pela aplicação da cinemática direta, onde pode-se obter a região de trabalho do robô chamada de "look up table". As coordenadas X,Y do robô são convertidas em ângulos para servomotores, que na realidade são valores que para cada servomotor varia de 0 a 255, onde cada valor vale aproximadamente 0,36 graus conforme descrito anteriormente. A posição dos ângulos dos servo-motores será enviada para

a porta serial do microcomputador que se comunica com o microprocessador do robô, o "mini-SSC". Ocorre então a transformação da posição dos ângulos do robô para "PWM" (modulação de largura de pulso), onde os pulsos controlados a uma determinada tensão são enviados para os servomotores do robô possibilitando-o se deslocar para a posição desejada. A Figura 5 ilustra este processo.

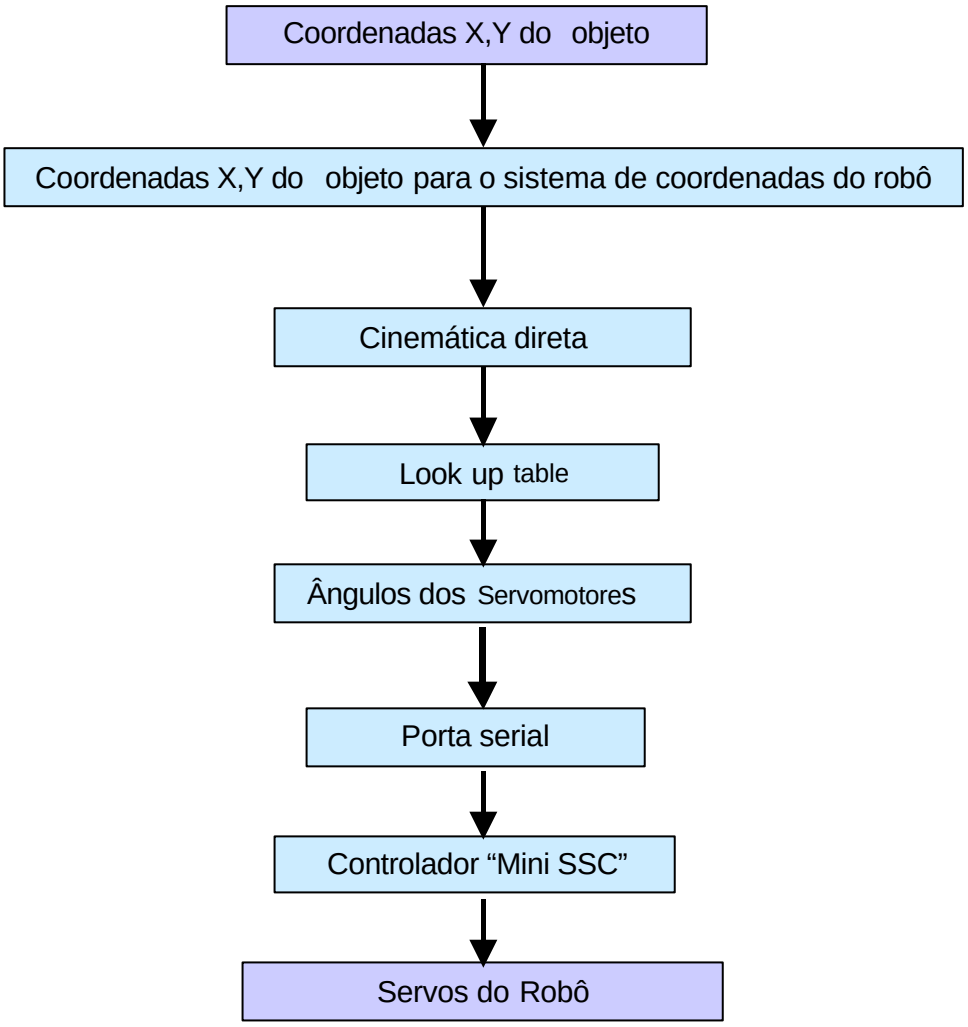

Figura 5 – Movimentação do Manipulador Robótico

4.3. Visualização Gráfica do Programa Desenvolvido

A seguir a visualização gráfica do programa, bem como a explicação de cada parte do mesmo. Como o programa foi construído com a linguagem visual C^{++}, fez-se uma interface gráfica (ilustrada na Figura 6) que permite ao usuário, além de colocar objetos para serem capturados pelo manipulador, poder fazer passo a passo as operações de captura de imagem, o cálculo do centróide, bem como a movimentação livre do robô pelo controle individual dos seus servomotores.

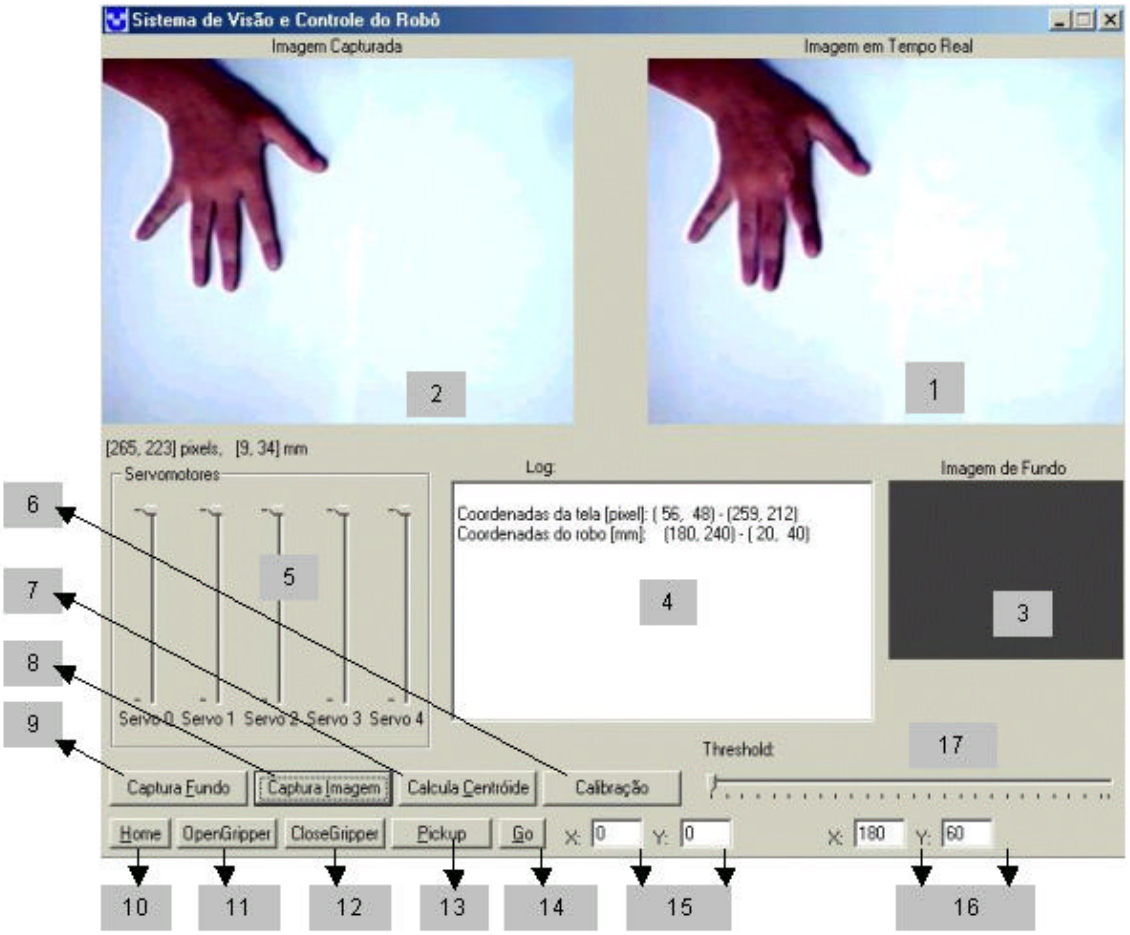

Figura 6 - Visualização gráfica da tela do programa.

Numerou-se de 1 a 17 as principais partes do programa, onde descreveu-se cada uma delas, e em seguida realizou-se uma simulação seqüencial do funcionamento do programa, com um exemplo que será dado a seguir.

1. Imagem em tempo real: é a cena que está sendo vista pela câmera.
2. Imagem capturada: é a imagem da figura a qual se deseja achar o centro de massa em coordenadas cartesianas.
3. Imagem de fundo: é a imagem que está no campo de ação visual antes do objeto ser colocado na cena.
4. Tela em Branco: mostra as coordenadas da tela em pixels, em mm e as coordenadas do robô em mm.
5. Servomotores: esta tela mostra o controle manual dos cinco servomotores.
6. Calibração: este botão é utilizado para calibração do sistema estabelecendo uma relação entre os sistemas de coordenadas da tela e o sistema real.
7. Calcula Centróide: este botão quando pressionado pelo usuário aciona o código do programa que encontra o valor numérico do centróide.
8. Captura imagem: botão que serve para capturar a imagem da cena.
9. Captura Fundo: este botão captura o fundo inicial do sistema, isto quer dizer que mesmo se for mudado a cor ou o relevo do fundo, o sistema consegue notar esta diferença e fazer a compensação.
10. Home: botão que ao ser pressionado leva o robô à posição inicial de trabalho.
11. “Open_Gripper”: ao ser pressionado abre totalmente a garra do robô.
12. “Close_Gripper” : ao ser pressionado fecha totalmente a garra do robô.

13. “Pick_up” : ao ser pressionado, permite que o robô execute a rotina de pegar a peça na região definida pelo sistema de visão, e colocar a mesma em um cesto conforme a programação.
14. “Go”: botão que ao ser pressionado leva o robô às coordenadas que podem ser definidas pelo usuário independente do sistema de visão.
15. X,Y sistema: posição x e y desenvolvida para colocar os valores numéricos calculados pelo botão. Calcula centróide em mm.
16. X,Y do robô: posição x e y desenvolvida para colocar os valores numéricos do eixo de coordenadas do robô em mm.
17. “Threshold”: esta barra de rolagem permite o ajuste manual do limiar.

4.3. Seqüência de Operação do Programa

Inicialmente, é necessário fazer a calibração do sistema, para isso faz-se a conversão do sistema de coordenadas, para que o programa possa alterar o sistema de coordenadas de “pixels” para mm. Para isso é necessário seguir os dados de entrada requeridos pelo sistema. No início do procedimento de calibração é necessário limpar a área de trabalho. Em seguida, deve-se colocar o objeto na posição inicial do sistema de coordenadas do robô, e clicar “ok” com o mouse ou apertar a tecla “enter” do PC. Então o sistema calcula a coordenada x,y do primeiro calibre utilizado. Este procedimento se repete quando o calibre é colocado no extremo do sistema em uma coordenada conhecida. Com posse dessas duas coordenadas calculadas, o sistema calcula a relação de pixel para mm, e consegue reconhecer o sistema de coordenada real de atuação do robô.

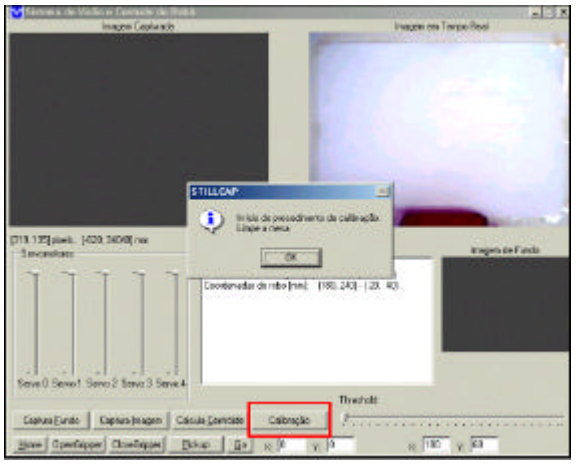

Figura 7 - Calibração do Sistema

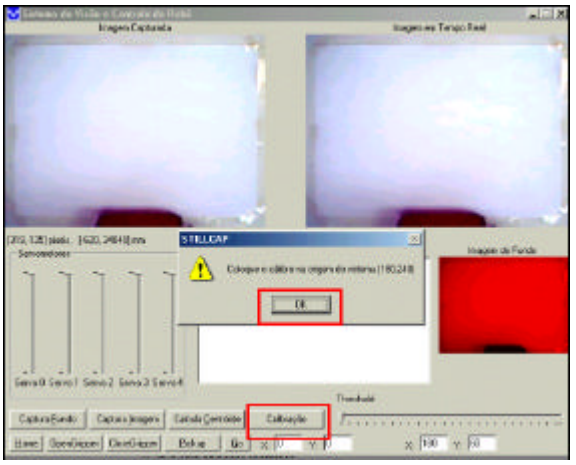

Figura 8 – Continuação da Calibração

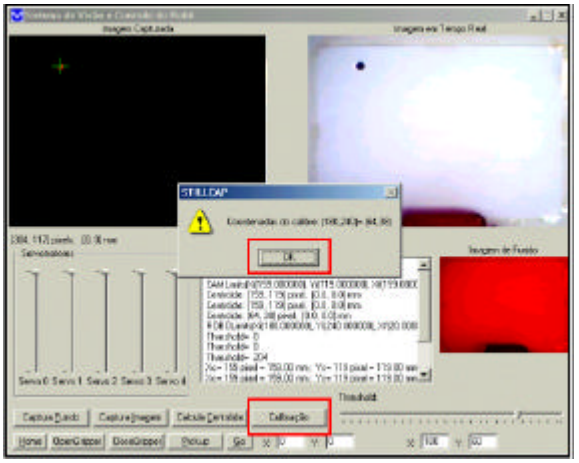

Figura 9 – Finalização da Calibração do Sistema

4.3.1. Capturando a Imagem no Campo de Visão da Câmera

Ao clicar na tecla ***captura o fundo*** será adquirida a imagem de fundo a ser utilizada no momento em que a peça é capturada pelo manipulador. Nota-se que foi colocado o cesto de depósito onde o robô colocará a peça. Uma vez que o sistema capturou este novo fundo, agora ele só "enxergará" a peça que será colocada posteriormente, conforme ilustrado na Figura 10.

Quando a peça é introduzida no sistema e o botão captura imagem é pressionado, conforme ilustrado na Figura 11.

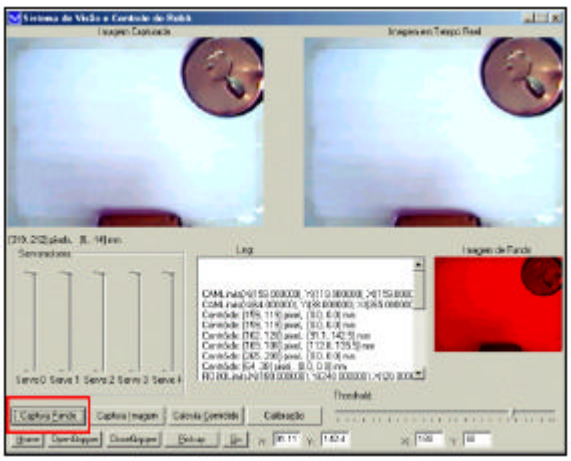

Figura 10 - Captura fundo

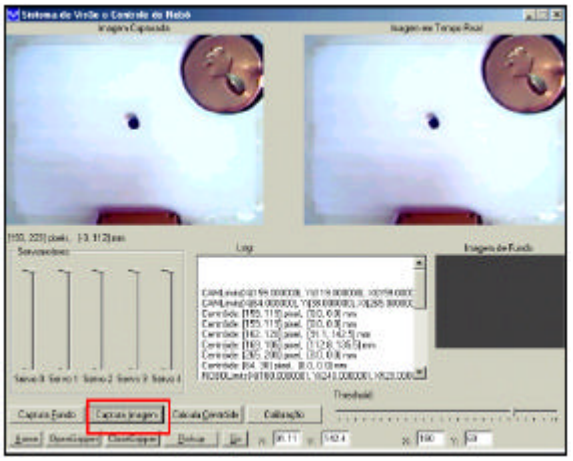

Figura 11 - Captura da Imagem

4.3.2. Calculando a Centróide da Imagem do Objeto

Ao ser pressionado o botão ***calcula centróide***, o programa calcula as coordenadas X,Y da peça, enviando os dados para a memória do programa, conforme ilustrado na Figura 12.

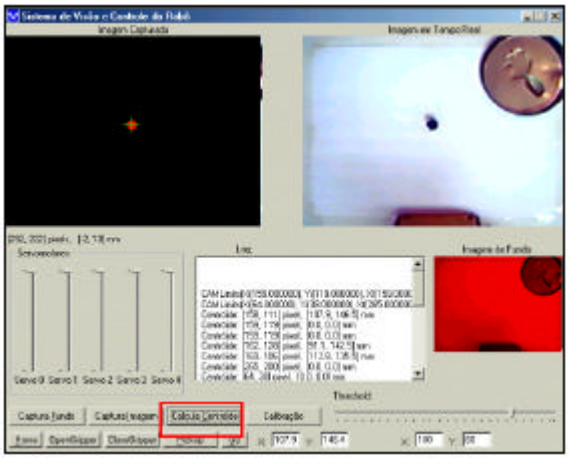

Figura 12 - Calcula Centróide

4.3.3. Comando "Pick Up"

Ao ser pressionada tecla ***pickup***, o manipulador retira os dados da posição da peça que estavam na memória e executa uma rotina que está em sua programação, que é pegar a peça e levá-la para um lugar aonde será depositada. Após a execução da tarefa pelo manipulador, ele retorna à posição de repouso ficando fora do campo visual do sistema de visão, conforme ilustrado nas Figuras 13 e 14. A Figura 15 ilustra o manipulador depositando a peça no recipiente desejado.

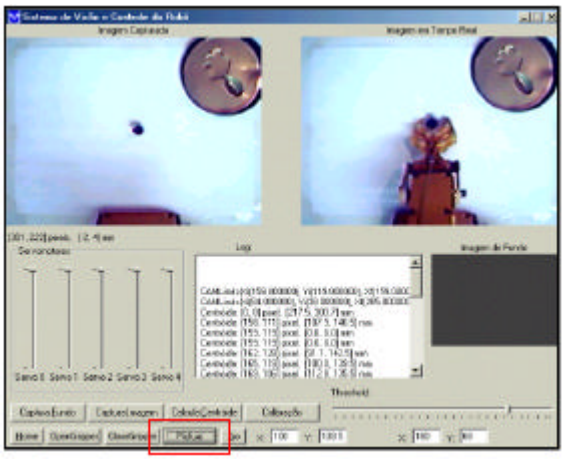

Figura 13 - Comando “Pick up

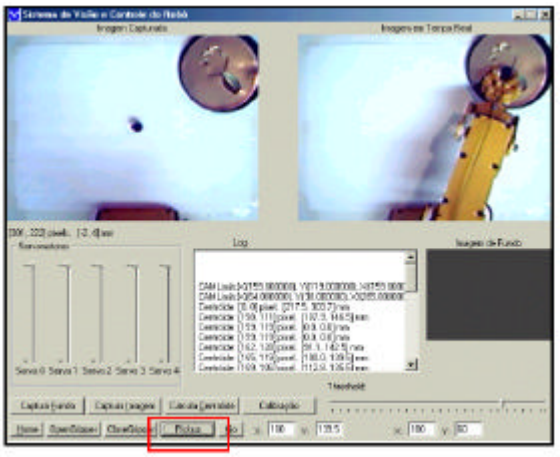

Figura 14 - Robô Depositando a Peça

Após o depósito da peça, o manipulador retorna à posição ***Home*** (posição inicial), e fica aguardando o próximo evento.

5. RESULTADOS E CONCLUSÕES

O trabalho começou a ser desenvolvido em linguagem C^{++} versão 3.1 da Borland, onde se programou o robô, independente do sistema de visão. Seria utilizada uma placa de captura de vídeo (frame grabber) do fabricante “Humusoft” devido a sua utilização em diversos outros trabalhos de visão robótica, o que possibilitava acesso a um grande material de pesquisa já disponível. Esta hipótese foi descartada devido a esta placa estar em desuso, e que por trabalhar no sistema operacional “DOS”, possui limitações de memória, o que poderia prejudicar o funcionamento do projeto em sua finalização.

Foi encontrado como alternativa desenvolver o restante da programação em linguagem Visual C^{++} versão 6.0 da Microsoft, que com sua linguagem visual suportou a migração de parte do programa que já havia sido desenvolvido, com algumas pequenas mudanças no seu código fonte. Outra vantagem oferecida por este software é a elaboração de um programa de captura de imagens que dispensaria o uso de uma placa específica de captura, podendo-se utilizar como sistema de visão um *web cam* ligada à entrada USB do microcomputador, ou também utilizar uma placa de captura com uma câmera e com o “driver” do fabricante.

Outra vantagem deste programa é que permite acessar as imagens através das bibliotecas do *Direct Show*, que são as rotinas utilizadas no programa na captura da imagem.

Verificou-se também, que o projeto satisfez a condição de um aplicativo de fácil compreensão, e que o usuário não necessita de conhecimento de programação para operar o programa devido ao seu caráter intuitivo. Nota-se também, que o robô utilizado conseguiu realizar sua tarefa de manipulação do objeto colocado em seu campo de trabalho com eficiência, sendo isto difícil de conseguir quando se trabalha com um robô que utiliza motores “DC” com malha aberta. Mas observou-se que com a metodologia aplicada ao robô, ele conseguiu efetuar sua tarefa de pegar e colocar um objeto sem cometer falhas.

6. REFERÊNCIAS

Kabayama, A. M. and Trabasso, L. G., 2002, “Performance evaluation of 3D computer vision techniques.”, *J. Braz. Soc. Mech. Sci.*, July 2002, vol.24, no.3, p.234-238., ISSN 0100-7386.

Motta, J. M. S. T. and McMaster, R. S., 2002, “Experimental validation of a 3-D vision-based measurement system applied to robot calibration.”, *J. Braz. Soc. Mech. Sci.*, July 2002, vol.24, no.3, p.220-225. ISSN 0100-7386.

Ferreira, M. F. O reconhecimento de Padrões. Dissertação de Mestrado. Universidade de Brasília, 1994.

Khatib,O. Real-time Obstacle Avoidance for Manipulators and Mobile Robots, Int'l Journal of Robotic Research, vol. 5, no.1, p. 90-98, 1986.

Querido, S. C. F. Implementação de um sistema de aquisição de imagem. Taubaté, 1999. Trabalho de graduação inter-disciplinar. p.67

Singh, J. Sanjiv and Meghanad Wagh, Robot Path Planning Using Intersecting Convex Shapes: Analysis and Simulation, IEEE Journal of Robotics and Automation, vol. RA-3, n.º 2, 1987.

Tou, J. T., Gonzales, R. C. Pattern Recognition Pinciples. Addilson-Wesley Publishing Company, Massachusetts, 1981.

Rillo, A. H. R. C., Bianchi, A. C., Moreira Jr., B. M.,Ferraz F., 1996, "Análise de Desempenho de Técnicas de Visão Computacional", Proceedings of the 11th Brazilian Automatic Control Conference, Vol. 1, São Paulo, S.P., Brazil, pp. 573-578.

Stemmer, M. R., Neto, C. A. M., 1998, "Reconhecimento de Imagens Utilizando Descritores de Fourier e Redes Neurais", Proceedings of the 12th Brazilian Automatic Control Conference, Vol. VI, Uberlândia, M.G., Brazil, pp. 2081-2085.

Mini, R. A. F., Campos, M. F. M., 1998, "Avaliação Automática do Volume do Ventrículo Esquerdo por Meio de Visão Computacional", Proceedings of the 12th Brazilian Automatic Control Conference, Vol. II, Uberlândia, M.G., Brazil, pp. 405-410.

Brown, L. J., Meyn, S. P., Weber, R. A., 1998, "Adaptive Dead-Time Compensation with Application to a Robotic Welding System", IEEE Transactions on Control System Technology, Vol. 6, No. 3, pp. 335-349.

Myler, H. R., Weeks, A. R, 1993, " Computer Imaging Recipes in C', Prentice Hall, New Jersey, 1993, 284p.

Lindley, C. A., Practical Image Processing in C - Acquisition, Manipulation, Storage, John Wiley & Sons, Inc., New York, 1991, 554p.

Rodrigues, D.L.,1997, "Procedimento automático para calibração de sistemas de visão robótica para operações *pick-and-place"*, São José dos Campos, Dissertação de Mestrado, Instituto Tecnológico de Aeronáutica, 126p.

Ribeiro, N. F. and Yamaguchi, O.,1999, "Cinemática Direta e Inversa de Manipuladores Robóticos com Elos Rígidos", Taubaté, S.P., Trabalho de Graduação Interdisciplinar, Departamento de Informática, Universidade de Taubaté. Parker, J. R., 1997, "Algorithms for Image Processing and Computer Vision", Wiley Computer Publishing,, USA, 417p.

< http://www.lynxmotion.com >(site para o kit do robô) consultado em 20/01/2003.

< http://www.webcam32.com > (site da câmera de vídeo utilizada) consultado em 30/01/2003.

< http://www.ontrak.net/mfc.htm > **(**Sending Commands in Visual C^{++} With MFC to ADR Interface) Consultado em 15/01/2003.

7. DIREITOS AUTORAIS

Os autores são os únicos responsáveis pelo conteúdo do material impresso incluído no seu trabalho.

PROGRAMMING A ROBOT MANIPULATOR PICK AND PLACE TASK AIDED BY A VISION SYSTEM

Álvaro Manoel de Souza Soares
University of Taubaté – Mechanical Engineering Department
Rua Daniel Danelli, s/n – Jardim Morumbi
ZIP: 12060-440 – Taubaté - SP. Brazil.
e-mail: alvaro@mec.unitau.br

Valdeci Donizete Gonçalves
University of Taubaté – Mechanical Engineering Department
Rua Daniel Danelli, s/n – Jardim Morumbi
ZIP: 12060-440 – Taubaté - SP. Brazil.
e-mail: valdeci@horizon.com.br

ABSTRACT: *This paper shows a robot doing a pick and place task, aided by a vision system. An experimental set up was assembled, composed by a didactic Lynks Motion four degree of freedom Robot and a web cam. The robot work space was mapped using a look up table, in order to minimize position errors. The main idea is that the robot must be able to pick, a bolt for example, in any place of its workplace, aided by the vision system. On the workplace we have a basket, place in a known position and a bolt placed anywhere. The vision system acquires the scene image, and locates the bolt gravity center. With this information, the robot pick the bolt and place it on the basket. In order to decrease the experimental costs, a web cam connected to a computer USB port was used. A computer program, in visual C++ language, was developed to acquire and treat the scene images and to control the robot. The program implements the Image processing algorithms and the robot direct kinematics. The system accuracy and repeatability are discusses and the results shown that Robotics and Image Processing algorithms work quite well.*

Keywords : Robotics, Vision system, Image Processing.