

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

Renan Luís Foresti

Universidade Federal do Rio Grande do Sul – Departamento de Engenharia Mecânica. Sarmento Leite 425, 90050-170, Porto Alegre, RS. renan.foresti@ufrgs.br

Flávio José Lorini

Universidade Federal do Rio Grande do Sul – Departamento de Engenharia Mecânica. Sarmento Leite 425, 90050-170, Porto Alegre, RS. lorini@ufrgs.br

1. INTRODUÇÃO

O surgimento do robô na indústria foi justificado principalmente pela necessidade de automação do processo produtivo, confiabilidade nos serviços executados e pela substituição do homem em locais de alta periculosidade. Segundo a definição do *Robot Institute of America*, “Um robô é um manipulador reprogramável e multifuncional, projetado para manipular material, peças, ferramentas ou dispositivos especializados, através de movimentos variáveis programados para o desempenho de uma variedade de tarefas”, porém, é difícil conceber-se um manipulador reprogramável e multifuncional trabalhando isoladamente nos dias atuais. Com o avanço das áreas tecnológicas, fica evidente que o objetivo é a integração das máquinas, visando reduzir tempos de produção e conseqüentemente custos. Os robôs ganham então, “sentidos”. Além da percepção sensorial, passam a se comunicar com outras máquinas, o que lhes garante uma inteligência artificial, podendo controlar e tomar decisões sobre o processo, sem a necessidade de intervenção humana. Um dos mais complexos desses “sentidos” e talvez o mais importante, é a visão, devido à riqueza de informações que uma imagem pode conter. Trata-se então, “ensinar os robôs a enxergar”, criando um sistema de visão artificial, semelhante à dos seres humanos. Isto não é tão simples, embora o sistema de visão artificial consiga operar em praticamente todo o espectro de radiações eletromagnéticas e com altíssima velocidade de processamento, não possui a capacidade de trabalhar sob condições muito variadas (iluminação, realce, mudança de características, etc.). É geralmente inflexível, apresentando bom desempenho apenas nas condições em que foi projetado, impondo-se um adequado processamento digital de imagens.

2. PROCESSAMENTO DE IMAGEM DIGITAL

Uma imagem digital pode ser representada por uma função discreta de intensidade de luz $f(x, y)$, onde x e y representam as coordenadas espaciais e f o brilho proporcional em cada ponto (x, y) . Em imagens monocromáticas, f está relacionado com os níveis de cinza em cada ponto, de modo que se pode representar uma imagem digital por uma matriz, onde cada elemento corresponde a um ponto da imagem digital; e os valores de seus elementos, correspondem ao nível de cinza da imagem, Gonzalez⁽²⁾. Este trabalho tem o objetivo de manipular imagens monocromáticas, portanto, faz-se necessário converter o modelo RGB (*red, green, blue*) para intensidade luminosa. Segundo Staugaard⁽⁷⁾ a equação (1) fornece uma excelente aproximação para a intensidade de luz monocromática (I), para os respectivos valores das componentes de cor vermelha (R), verde (G) e azul (B) respectivamente.

$$\mathbf{I} = 0,3.\mathbf{R} + 0,59.\mathbf{G} + 0,11.\mathbf{B} \quad (1)$$

As principais técnica de análise de imagens tratadas e implementadas no aplicativo são resumidamente descritas na seqüência do texto.

2.1. Limiarização (*Thresholding*)

A limiarização é uma das mais importantes técnicas de segmentação de imagens. É através dela que se consegue separar os objetos do fundo de uma imagem. Em uma imagem monocromática, por exemplo, com dois conjuntos representando a conectividade entre os pixels, pode-se dizer que um grupo representa o objeto na imagem e o outro grupo representa o fundo. Sendo assim, os dois grupos foram separados por um *limiar* T , fazendo com que os conjuntos de conectividade assumam um novo valor, tornando a imagem monocromática em uma imagem binária. O *limiar* T pode ser entendido pela equação (2), onde $f(x, y)$ é a função que representa a intensidade luminosa na imagem, e $g(x, y)$, a nova função de luminosidade depois da separação do *limiar* T . Marques⁽⁵⁾

$$g(x, y) = \begin{cases} 1 & \text{se } f(x, y) > T \\ 0 & \text{se } f(x, y) \leq T \end{cases} \quad (2)$$

2.1.1. Método de Otsu

Este método, descrito em Gonzalez⁽³⁾, foi proposto por Otsu em 1979. Consiste em achar um limiar que maximize a variância σ^2 entre os dois grupos de pixels (fundo e objeto). Para a implementação do método, é necessário normalizar o histograma aplicando a Equação (3). Para um determinado *limiar* T , têm-se dois grupos de pixels, o grupo C_0 , formado pelos valores $\{0, 1, 2, \dots, T-1\}$ e o grupo C_1 , formado pelos valores $\{T, T+1, \dots, L-1\}$.

$$p_r(r_k) = \frac{n_k}{n} \quad (3)$$

onde: $0 \leq r_k \leq L-1$; $k = 0, 1, 2, \dots, L-1$, onde L é o número de níveis de cinza da imagem; n é o número total de pixels na imagem; $p_r(r_k)$ = percentual do k -ésimo nível de cinza e n_k = número de pixels cujo nível de cinza corresponde a k . Através da Equação (4) encontra-se a variância σ_B^2 .

$$\sigma_B^2 = w_0(\mu_0 - \mu_T)^2 + w_1(\mu_1 - \mu_T)^2 \quad (4)$$

onde, σ_B^2 é a variância, w_0 é a probabilidade do grupo C_0 (Equação 5), w_1 é a probabilidade do grupo C_1 (Equação 6), μ_0 é a média do grupo C_0 (Equação 7), μ_1 é a média do grupo C_1 (Equação 8), e μ_T é a média total do histograma (Equação 9). Na resolução da Equação (4) para todos possíveis limiares, seleciona-se aquele que resultar no maior valor de σ_B^2 .

$$w_0 = \sum_{k=0}^{T-1} p_r(r_k) \quad (5)$$

$$w_1 = \sum_{k=T}^{L-1} p_r(r_k) \quad (6)$$

$$\mu_0 = \sum_{k=0}^{T-1} k \frac{p_r(r_k)}{w_0} \quad (7)$$

$$\mu_1 = \sum_{k=T}^{L-1} k \frac{p_r(r_k)}{w_1} \quad (8)$$

$$\mu_T = \sum_{k=0}^{L-1} k p_r(r_k) \quad (9)$$

2.2. Morfologia Matemática

O princípio básico da morfologia matemática consiste em extrair as informações relativas à geometria e topologia de um conjunto desconhecido (uma imagem) pela transformação através de outro conjunto completamente definido, chamado de *elemento estruturante*. Portanto, a base da morfologia matemática está estruturada na Teoria de Conjuntos. Hirata⁽⁴⁾.

3. COMPONENTES DO SISTEMA IMPLEMENTADO

3.1. Dispositivo para transmissão de dados

A transmissão dos dados resultante do processamento de imagem ocorre através de um dispositivo de hardware dedicado, o *IRBCom* desenvolvido por Bayer⁽¹⁾, que permite comunicação bidirecional entre o robô ABB IRB1400 e um microcomputador. O dispositivo integra a porta paralela do microcomputador a uma placa de comunicação digital com 16 canais de entrada e 16 de saída para sinais discretos, originalmente destinada a integração do controle do robô a dispositivos externos. O dispositivo opera via software através de rotinas compiladas em formato *dll* (*Dinamic Library Link*).

3.2. Robô

O manipulador utilizado corresponde a um robô *IRB1400* da ABB, um equipamento dotado de seis graus de liberdade com sistema operacional e linguagem de programação proprietários.

3.3. Câmeras

A instalação utiliza uma câmera CCD *SAMSUNG BW-2302ED* com resolução de 720x480 pixels. Testes comparativos também foram realizados com uma *webcam Genius VideoCAM Express V2* (resolução 352x288 pixels).

4. ROTINAS DE PROGRAMA

O desenvolvimento de rotinas de software para processamento de imagens utiliza recursos implementados no aplicativo *MATLAB versão R2006a* e *Microsoft Visual Studio 2005*.

4.1. Funções *MATLAB*

O *MATLAB*, *MATrix LABoratory*, é um software programável da “*The MathWorks*”. O *MATLAB* apresenta bibliotecas de rotinas específicas (*toolbox*) para cada aplicação, inclusive para o processamento de imagens. Todas as funções pré-definidas em uma biblioteca podem ser modificadas, isto significa que é possível escrever funções específicas para a adequação de cada problema em particular. Uma característica importante e útil do *MATLAB*, é que os programas escritos podem ser transformados em programas executáveis independentes do ambiente *MATLAB* utilizando-se o “*MATLAB Compiler*”, Matsumoto⁽⁶⁾.

4.2. Função *Boundary Lines*

Com o objetivo de tornar o processamento automático e simples para os usuários, foi desenvolvido um software de controle, em *Visual Basic .NET*, chamado *Boundary Lines*, responsável pelo controle do *MATLAB*, manipulação dos resultados, geração da programação para o robô (em código *RAPID*) e envio de dados para o *IRBCom*.

5. CALIBRAÇÃO

Para evitar erros de distorção da imagem, tanto a *webcam* quanto a câmera CCD foram posicionadas perpendicularmente ao plano de trabalho do robô, de tal forma que o plano da imagem pertence ao plano do trabalho. Uma correção de escala fez-se necessária (Equação 10), pois o *MATLAB* retorna a posição (x, y) medida em pixels e o robô requer as coordenadas (X, Y) em milímetros.

$$F(X, Y) = \alpha \cdot f(x, y) \quad (10)$$

onde $F(X, Y)$ é a matriz que representa o plano de trabalho, $f(x, y)$ é a matriz que representa o plano da imagem e α é o fator de correção que representa o comprimento de uns dos lados do pixel em milímetros, considerando o pixel com razão 1/1 de altura e largura. Conhecida a distância focal e o tamanho do plano da imagem virtual das câmeras, utilizam-se facilmente as Leis da Ótica para definir o valor de α . Caso isso não seja possível, pode-se utilizar uma aproximação prática experimental, que consiste em obter uma imagem de um objeto de tamanho conhecido e relacionar dimensões ao número de pixels.

Dependendo da aplicação, muitas vezes a imagem é adquirida em uma posição diferente do plano de trabalho, caso de planos (imagem e trabalho) transladados. Neste trabalho utilizaram-se os planos coincidentes e transladados.

6. TESTES E AVALIAÇÃO DO SISTEMA

A definição de um objeto claro é relativa, pois dependente do fundo, um objeto é considerado claro se o seu nível de cinza for maior que o do fundo, mas o mesmo objeto pode ser considerado escuro se o seu nível de cinza for menor que o do fundo. Na implementação do sistema o *Boundary Lines*, utilizando o *MATLAB*, diferencia automaticamente o fundo do objeto. Dois distintos objetos, testados para avaliação do sistema em condições de diferentes contrastes (fundo-peça), são ilustrados na Figura 1.

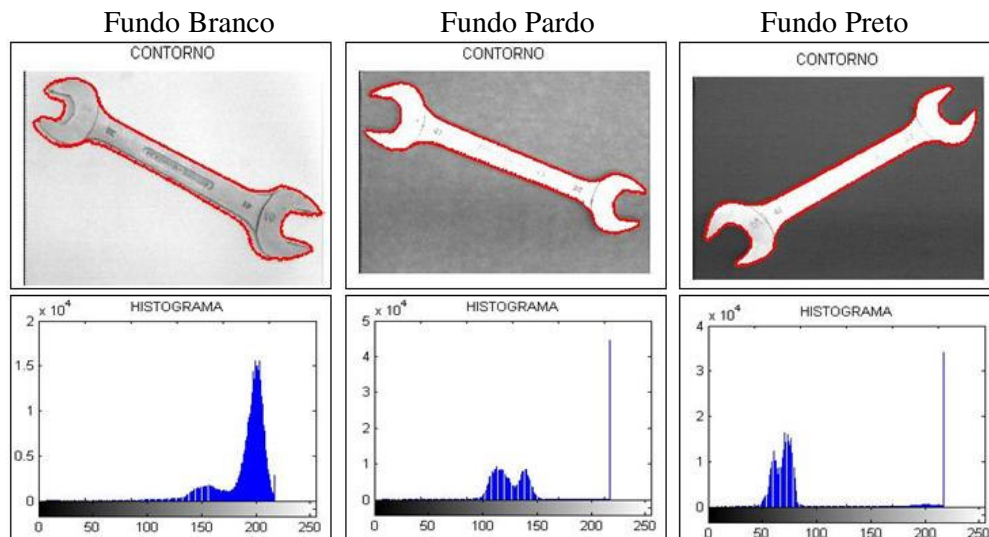


Figura 1 – Resultado do processo de limiarização

O erro de contorno mostra-se mais acentuado quando utilizado o fundo branco, causado pela influência da iluminação. Pode-se notar no histograma do objeto de fundo branco, Figura 1, que não há uma separação visível entre os *pixels*.

As trajetórias obtidas são enviadas via *IRBCom* para o controle do robô. O robô, por sua vez, reproduz as trajetórias - Figura 2.

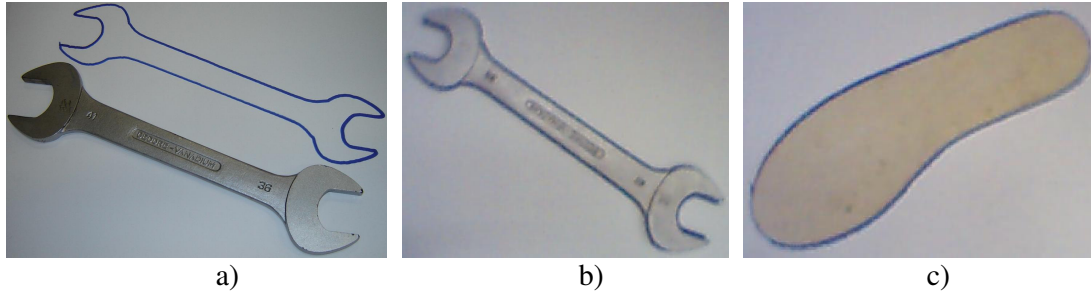


Figura 2 – Imagens capturadas e reprodução de contornos

A precisão de trajetória reproduzida em função da captura ser realizada por câmera CCD ou *webcam* é ilustrada na Figura 3, destacando-se uma melhor precisão quando a câmera CCD é utilizada. A imagem gerada pela *webcam*, devido a um número menor de pontos, produz trajetórias mais rápidas comparadas com imagens geradas pela câmera CCD, desta maneira, pode-se justificar o uso de um *webcam* quando a precisão do contorno for relevante.

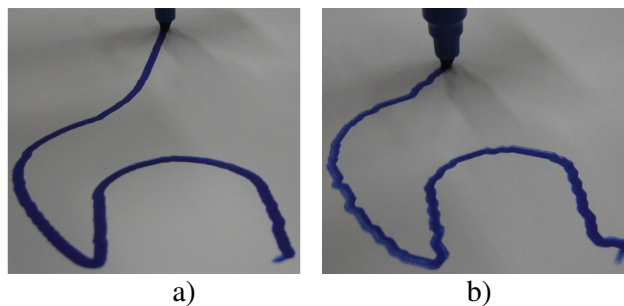


Figura 3 – Imagem capturada por câmera CCD (a) e *webcam* (b)

7. CONCLUSÕES

O Sistema de Visão proposto atende aos objetivos deste trabalho. O sistema demonstrou confiabilidade em encontrar o contorno de objetos posicionados de forma diferentes, variando-se o contraste da imagem, e no envio e execução das trajetórias. Com exceção da calibração inicial, o sistema demonstrou autonomia e independência de ajustes ou intervenção humana durante o ciclo de trabalho.

O uso da câmera CCD tem a vantagem de preservar o sinal analógico da imagem. Para a *webcam*, tem-se a vantagem de não necessitar de uma placa de aquisição de imagens. Modelos de *webcam* com resoluções semelhantes à câmera CCD, poderiam eliminar o problema das trajetórias mostrado na Figura 3 (b), justificando-se sua viabilidade.

O envio das trajetórias via *IRBCom* é relativamente lento, possuindo um ciclo de operação de aproximadamente 1,5 s para cada ponto, tornando-se característica crítica do sistema. Esse resultado já era esperado com o *IRBCom*, que obteve resultados excelentes em apenas operações *pick-and-place* (Bayer, 2004). Uma solução para maiores taxas de transmissão seria possível via conexão serial ou barramento de rede, opções não disponíveis para o controle do robô utilizado.

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

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

8. REFERÊNCIAS

1. BAYER, F. M. **Desenvolvimento e Implementação de uma Interface de Comunicação para Controlador Robótico Industrial**. 2004. 87 f. Dissertação (Mestrado em Engenharia Mecânica), Universidade Federal do Rio Grande do Sul, Porto Alegre, 2004.
2. GONZALEZ, R.C.; WOODS, R.E. **Processamento de Imagens Digitais**. 1. ed. São Paulo: Edgard Blücher, 2000. 509 p.
3. GONZALEZ, R.C.; WOODS, R.E.; EDDINS, S.L. **Digital Image Processing Using MATLAB**. 1. ed. Upper Saddle River: Pearson Prentice Hall, 2004. 609 p.
4. HIRATA, N. S. T. **Noções de Morfologia Matemática.pdf** São Paulo, 22 maio 2002. 1 arquivo (420 bytes). Adobe Acrobat 7.0 Document.
5. MARQUES Filho, O.; VIEIRA Neto, H. **Processamento Digital de Imagens**. 1. ed. Rio de Janeiro: Brasport, 1999. 409 p.
6. MATSUMOTO, E.Y. **MATLAB 6.5 Fundamentos de Programação**. 2. ed. São Paulo: Érica Ltda, 2004. 432 p.
7. STAUGAARD, A. C. In: Robotics and AI – An Introduction to Applied Machine Intelligence. **Vision**. Englewood Cliffs, 1987. p 196-265.