

Correlação de Imagens Coloridas Visando Auxiliar na Navegação e Controle de Robôs Autônomos

Eduardo Maestri Righes¹, Fernando Santos Osório¹

¹Pós-Graduação em Computação Aplicada – PIPCA
UNISINOS – São Leopoldo, RS

eduardo.righes@gmail.com, fosorio@unisininos.br

Abstract. *Today, autonomous mobile robots are growing their importance in Artificial Intelligence research development. In this field, there are relevant research topics like: maintenance of the robot position, path planning and navigation. In order to implement this kind of tasks, the use of image acquisition systems are increasing. This work presents a new technique to visual robot navigation based on color image correlation. We also discuss about the results we obtained in experiments using real images and the possible application of the proposed method in robot control tasks.*

Resumo. *Atualmente a robótica autônoma móvel vem assumindo um importante papel no desenvolvimento de pesquisas em Inteligência Artificial. Nesta área existem tópicos que são de fundamental importância, a saber: manutenção da localização do robô, planejamento de trajetórias e navegação. Um dos dispositivos sensores que vem sendo cada vez mais usados em robótica móvel são os sistemas de aquisição de imagens (câmeras de vídeo). Este trabalho se propõe a apresentar uma nova técnica para navegação visual baseada na correlação entre imagens coloridas. São discutidos os resultados de experimentos realizados com imagens reais e a sua aplicação no controle de um robô móvel.*

1. Introdução

Os robôs móveis [Dudek and Jenkin, 2000] são dispositivos que buscam perceber o ambiente em que estão inseridos, planejar e realizar ações. Busca-se dotar estes robôs de um controle autônomo [Heinen, 2002] responsável pela manutenção da localização do robô, planejamento de trajetórias e navegação, o que é feito a partir da leitura e interpretação dos dados dos sensores. O uso de sistemas de aquisição de imagens (câmeras de vídeo) vem se destacando como forma de obter estas informações do ambiente, permitindo uma adequada execução das tarefas descritas acima.

Este trabalho se propõe a utilizar uma técnica similar à apresentada em [Jones et al., 1997], que utiliza *normalized cross-correlation* (NCC) [Martin and Crowley, 1995] para fazer o *matching* entre duas imagens de modo a obter uma referência relativa à posição atual do robô que possui uma câmera acoplada. Entretanto, em [Jones et al., 1997] somente é abordada a correlação baseada em imagens monocromáticas do tipo *grayscale*. Imagens coloridas proporcionam um volume de informação e uma riqueza de detalhes maior do que imagens em *grayscale* [Gonzalez and Woods, 2002]. Deste modo, buscamos adaptar as técnicas de processamento de imagens de modo a aproveitar a informação que as cores podem proporcionar.

2. Trabalhos Relacionados

Atualmente a robótica móvel é empregada em diversos campos, desde a automação industrial até a exploração espacial [Dudek and Jenkin, 2000]. Na robótica móvel autônoma, a navegação do robô pode ser feita através de marcas inseridas no ambiente, pela odometria ou a partir de informações visuais, obtidas por uma câmera acoplada ao robô [Matsumoto et al., 1996, Heinen, 2002, Jung et al., 2005].

Várias propostas são encontradas na literatura [Matsumoto et al., 1996, Jones et al., 1997, Hu and Uchimura, 2002, Matsumoto et al., 1999, Gross et al., 2003], descrevendo sistemas que permitem uma navegação e controle do robô através do uso de visão computacional. Estes sistemas são usualmente denominados de *sistemas de navegação visual* [Jung et al., 2005]. O objetivo de alguns destes sistemas é imitar o comportamento humano, onde uma pessoa é capaz de ser conduzida por um

caminho, armazenar em sua memória o caminho percorrido, e em um momento posterior, realizar de modo autônomo novamente este mesmo caminho, baseando-se em suas lembranças. Dentre os trabalhos pesquisados, destaca-se o de [Jones et al., 1997], que utiliza este tipo de técnica. Neste trabalho optamos em seguir um modelo similar ao apresentado por [Jones et al., 1997].

Em [Jones et al., 1997], a NCC é utilizada para determinar o posicionamento atual de um robô. O robô captura uma imagem, e faz o cálculo de correlação contra imagens de referência previamente armazenadas. Estas imagens de referência compõem um caminho ou trajetória que o robô deverá seguir. Primeiramente determinamos o desvio de orientação em relação a imagem alvo, corrigindo através de comandos de rotação do robô até que a imagem esteja melhor "enquadrada". Depois, conforme o robô avança na sua trajetória, o campo de visão irá se alterar, logo o valor da correlação vai diminuindo. Quando não satisfizer mais um determinado limite, uma nova imagem de referência é adotada, sendo usualmente esta imagem a seguinte na seqüência que descreve o caminho, e o processo então se repete: ajuste de orientação e avanço.

O estudo que [Jones et al., 1997] realizou utilizava imagens *grayscale*. Neste trabalho, visamos o estudo de correlação em imagens coloridas onde adotamos o espaço de cores RGB (Red-Green-Blue) [Gonzalez and Woods, 2002]. A escolha deste espaço de cores foi baseada principalmente pela sua simplicidade. Pretendemos em trabalhos futuros realizar estudos em outros espaços de cores, tais como o HSV/HSI. Para fazer a correlação em imagens coloridas, adotamos uma técnica que avaliou individualmente cada componente de uma imagem RGB, que depois foram "integradas" (juntadas) através de um método baseado em vetores e correlação, como descrito na Seção 3.

3. Vetores de Correlação em Imagens Coloridas

Para determinar uma possível mudança na trajetória do robô, precisamos definir uma forma de informar ao robô sobre o seu desvio em relação as imagens de referência para que este corrija sua trajetória. Desta forma, precisamos definir um mecanismo para que o robô possa determinar seu deslocamento relativo a um ou mais pontos de referência sendo utilizados em um dado momento.

O conjunto de imagens que o robô utilizará como referência para seguir uma determinada trajetória é chamado de banco de imagens (BI). Cada entrada do BI contém uma imagem de referência (IR_i) e os pontos de referência ($P_{ref_{ij}}$), que são subimagens de IR_i . O BI pode possuir n entradas, onde i varia de $1..n$. Uma entrada i do BI pode possuir m subimagens de IR_i ($P_{ref_{ij}}$), onde j varia de $1..m$. As imagens que o robô captura são denominadas ICR – Imagem Capturada pelo Robô. O referencial usado para determinar os deslocamentos, tanto nas ICR quanto nas IR_i será o ponto central de cada imagem (C_x, C_y). Este ponto também é chamado de P_{alvo} quando estamos nos referenciando a uma ICR. O ponto (C_x, C_y) é utilizado como referência por que a correlação entre duas imagens idênticas tem seu valor máximo neste ponto. Para determinar (C_x, C_y), assumimos que a imagem tem x colunas por y linhas, sendo o ponto central dado por $(\frac{x}{2}, \frac{y}{2})$. Quando nos referirmos a um componente de uma determinada imagem, ele será indexado pelo índice c . Por exemplo, ICR_c está referenciando o componente c da ICR sendo analisada. De posse das definições básicas, podemos definir como é atualizada a posição do robô em relação a uma IR_i do BI.

O robô corrige sua trajetória seguindo o processo descrito a seguir. Primeiramente, precisamos de uma referência para guiar o robô. Assim, recuperamos do BI uma determinada entrada i . Essa entrada consiste da IR_i e uma ou mais subimagens $P_{ref_{ij}}$. Com estes dados, podemos calcular, para cada componente do espaço RGB, a posição de cada $P_{ref_{ij}}$ na IR_i . Ou seja, o quão deslocados estão os pontos de referência do ponto (C_x, C_y) da IR. Isso pode ser visto no algoritmo 1.

O algoritmo NCC [Martin and Crowley, 1995] obtém a partir de duas imagens (contendo apenas um componente) uma matriz M bidimensional com valores normalizados entre [-1,1] que indicam o grau de correlação entre as imagens. Os valores da correlação são obtidos através do deslizamento de uma janela ($P_{ref_{ijc}}$) sobre uma imagem de referência (IR_{ic}) e, para cada posição da janela, é aplicado o algoritmo NCC.

O vetor D_{ref} é composto da posição (par de coordenadas X,Y) onde a correlação alcançou seu valor máximo e do valor deste pico de correlação, isso para cada componente do espaço RGB. As posições dos picos de correlação são os deslocamentos das subimagens $P_{ref_{ij}}$ relativos ao ponto (C_x, C_y) da IR. As posições dos picos de correlação encontrados na matriz M devem ser mapeadas para

Entrada: Uma imagem de referência IR_i e um conjunto de subimagens $P_{ref_{ij}}$

Saída: Um vetor D_{ref} composto por x , y e valor máximo de correlação

$i \leftarrow$ Entrada Atual do BI ;

para todo $j \leftarrow 1$ **até** m **faça**

para cada Componente c de RGB **faça**

$M \leftarrow \text{NCC}(P_{ref_{ijc}}, IR_{ic})$;

$D_{ref_j} \leftarrow \text{CalculaDeslocamento}(M)$;

fim

fim

1: Algoritmo - Cálculo dos deslocamentos das subimagens $P_{ref_{ij}}$ em relação ao centro da imagem de referência IR_i

a IR_i . Isso é feito através da rotina `calculaDeslocamento()`, onde as dimensões da matriz M são mapeadas para as dimensões de IR_i . Feito isto, deve ser então realizado o procedimento equivalente para a ICR, onde ao invés de informarmos uma IR_i informamos a ICR, obtendo como resultado um vetor D_{alvo} .

Assim como D_{ref} , D_{alvo} possui os deslocamentos das subimagens $P_{ref_{ij}}$ relativos a P_{alvo} e o valor do pico de correlação. Com estes valores, D_{ref} e D_{alvo} , podemos gerar vetores para avaliarmos o deslocamento relativo entre a ICR e a IR_i para cada componente da imagem. A partir dos resultados obtidos, são traçados os vetores referentes ao processamento da subimagem $P_{ref_{i1}}$. Com isso, podemos fazer uma análise para determinar se os valores de cada um dos três componentes (RGB) da imagem obtidos estão convergindo, ou seja, apontam para a mesma direção. Para determinar se existe a convergência ou não, definimos uma distância limite $Limite_V$ entre os vetores V_r , V_g e V_b , gerados a partir de uma entrada j de D_{ref_j} ou D_{alvo_j} e estes não podem violar tal limite. É calculada a distância euclidiana entre os vetores V_r , V_g e V_b e caso algum não satisfaça o $Limite_V$, o *matching* é rejeitado. Isto pode ser visto nas equações 1, 2 e 3.

$$B_1 = \sqrt{(V_{rx} - V_{gx})^2 + (V_{ry} - V_{gy})^2} \leq 2.Limite_V \quad (1)$$

$$B_2 = \sqrt{(V_{rx} - V_{bx})^2 + (V_{ry} - V_{by})^2} \leq 2.Limite_V \quad (2)$$

$$B_3 = \sqrt{(V_{gx} - V_{bx})^2 + (V_{gy} - V_{by})^2} \leq 2.Limite_V \quad (3)$$

Se todos valores booleanos B_1 , B_2 e B_3 forem verdadeiros, o *matching* é aceito, caso contrário é rejeitado.

Acima foi discutido o limite $Limite_V$, relacionado a posição. Existe também um segundo limite, o $Limite_C$, que determina a qualidade da correlação. O $Limite_C$ é um limiar aplicado sobre o valor de correlação obtido, pois cada ponto da imagem terá um valor associado e este é normalizado entre $[-1, 1]$. Se o valor de correlação de um determinado vetor estiver abaixo de $Limite_C$, o *matching* também é rejeitado.

4. Experimentos

Os experimentos foram feitos utilizando-se imagens 24-bits 640x480 pixels. A implementação da *normalized cross-correlation* foi fornecida pelo *Toolbox* de processamento de imagens do Matlab [Mathworks, 2002], versão 6.1 R12. Foi realizada uma simulação da captura de imagens e do controle do robô (rotação e deslocamento). Foi gerado um banco de imagens (BI) em ambientes internos com uma iluminação controlada. Neste trabalho não foi considerado o desvio de obstáculos imprevistos (não presentes nas imagens originais). O algoritmo também assume que o robô inicia em uma posição que gere uma ICR semelhante a primeira entrada do BI.

O algoritmo utilizado para a simulação realiza a seguinte seqüência de passos, conforme descrito a seguir. Inicialmente, é capturada uma imagem (ICR) e, de posse dela, é realizado o cálculo de correlação das subimagens $P_{ref_{ij}}$ contra a IR_i e também contra a ICR. Depois é verificada a qualidade da correlação

obtida através do limite $Limite_C$. Caso algum valor de correlação esteja abaixo de $Limite_C$, a entrada i do BI é incrementada em 1. Se o critério $Limite_C$ for satisfeito, deve ser feita a análise dos vetores V_r , V_g e V_b obtidos para determinar se existe convergência entre eles (satisfazem as Equações 1, 2 e 3)

Para a geração do comando enviado ao robô, é feita a análise da diferença entre o vetor de uma $P_{ref_{ij}}$ obtido na IR e do vetor obtido com a mesma $P_{ref_{ij}}$ na ICR. Os comandos são enviados ao robô para que ele tente se posicionar de modo que os vetores das subimagens $P_{ref_{ij}}$ encontrados na ICR se movimentem até que se aproximem dos vetores obtidos com as mesmas subimagens na IR, ou seja, aproximando deste modo P_{alvo} da ICR do (C_x, C_y) da imagem IR_i . O robô terá conseguido se posicionar se os vetores obtidos na IR e na ICR satisfizerem o limite $Limite_V$. Conseguindo isso, o robô pode ser considerado posicionado com uma orientação adequada, pois está na trajetória correta e portanto está apto a avançar, mudando para a próxima imagem de referência.

No experimento realizado foi possível simular a navegação da posição inicial, com ajustes de orientação e posição, realizando o deslocamento até o destino, demonstrando o funcionamento adequado do método proposto.

5. Conclusões

Os experimentos realizados até o presente momento nesta pesquisa apresentaram bons resultados. O uso de cores e a análise da correlação através de vetores e pontos de referência pré-selecionados pode tornar a navegação de um robô autônomo mais robusta, no que diz respeito inclusive à modificações no ambiente e oclusões. As perspectivas para trabalhos futuros apontam para o tratamento de problemas relativos a disparidades muito grandes na iluminação das fotos. Existe a possibilidade da realização de testes em outros espaços de cores, como o HSI. E, por fim, a realização de estudos a fim de selecionar de forma automática subimagens $P_{ref_{ij}}$ que sejam representativas para a tarefa de navegação. Além disto pretende-se testar o sistema em um robô real.

Referências

- Dudek, G. and Jenkin, M. (2000). *Computational Principles of Mobile Robotics*. Cambridge University Press.
- Gonzalez, R. C. and Woods, R. E. (2002). *Digital Image Processing 2nd Edition*. Prentice Hall.
- Gross, H.-M., Koenig, A., Schroeter, C., and Boehme, H.-J. (2003). Omnivision-based probabilistic self-localization for a mobile shopping assistant continued. In *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, Nevada – USA.
- Heinen, F. J. (2002). *Sistema de Controle Híbrido para Robôs Móveis Autônomos*. Unisinos / PIPCA – Mestrado em Computação Aplicada, São Leopoldo – RS, Brasil. Dissertação de Mestrado.
- Hu, Z. and Uchimura, K. (2002). Dynamical road modeling and matching for direct visual navigation. In *Proceedings of the 6th IEEE Workshop on Applications of Computer Vision (WACV'02)*.
- Jones, S. D., Andersen, C. S., and Crowley, J. L. (1997). Appearance based processes for visual navigation. In *Proceedings of the 5th International Symposium on Intelligent Robotic Systems, SIRS'97*, pages 227–236, Stockholm, Sweden.
- Jung, C., Osório, F., Kelber, C., and Heinen, F. (2005). Computação embarcada: Projeto e implementação de veículos autônomos inteligentes. In *Anais do XXV Congresso da SBC - JAI: Jornadas de Atualização em Informática*, São Leopoldo, RS.
- Martin, J. and Crowley, J. L. (1995). Experimental comparison of correlation techniques. In *Proceedings of the International Conference on Intelligent Autonomous Systems*, Karlsruhe, Germany.
- Mathworks (2002). Acessado em <http://www.mathworks.com>.
- Matsumoto, Y., Ikeda, K., Inaba, M., and Inoue, H. (1999). Visual navigation using omnidirectional view sequence. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*.
- Matsumoto, Y., Inaba, M., and Inoue, H. (1996). Visual navigation using view-sequenced route representation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 83–88, Minneapolis, Minnesota.